National

パーソナルコンピュータ

MSX

DISK BASIC
MSX-DOS

使用説明書

National

MSX パーソナルコンピュータ

DISK BASIC/MSX-DOS説明書

はじめに

このたびはナショナルパーソナルコンピュータをお買い上げいただきまして，まことにありがとうございました．

本書はフロッピーディスクをメディア(媒体)として使用するための，MSX DISK BASICならびにMSX-DOSの文法書です．

本書では初心者の方にも活用していただけるように，親しみやすい使用例を随所に入れてあり，また，実用的なサンプルプログラムとその解説も用意しました．

なお，プログラム作成に使用するMSX BASICにつきましては，お手持ちのMSX BASIC説明書などをお読みください．

本書の使い方

本書は次のような構成になっています.

第1章　DISK BASICの概要

DISK BASICの特徴と，DISK BASICを使用するときに必要となる知識，およびファイルの入出力についてを説明しています.
これらは，第2章の命令の使い方についての説明を理解するために必要なことがらですが，初心者の方で，第1章を読むのが重荷であるときは，とりあえず第2章や第3章のサンプルプログラムを模倣したり改造してみたりすることから始めるとよいでしょう.

第2章　DISK BASICのコマンド・ステートメント・関数

DISK BASICで用意している命令(コマンドやステートメント）や関数の使い方をアルファベット順に説明しています.

第3章　サンプルプログラム

第2章で説明した命令をいくつか取り上げ，実践的なサンプルプログラムを紹介しながら解説しています.

第4章　MSX-DOSの概要

MSX-DOSの特徴と，MSX-DOSを使用するときに必要となる知識を説明しています.

第5章　MSX-DOSのコマンド

MSX-DOSで用意している命令の使い方をアルファベット順に説明しています.

第6章　資　料

メモリマップ，エラーメッセージ表などプログラムの作成に役立つ資料をまとめてあります.

ご　注　意

目　次

第１章　DISK BASICの概要

第２章　DISK BASICリファレンス

第３章　サンプルプログラム

第４章　MSX-DOSの概要

第５章　MSX-DOSリファレンス

第６章　資料編

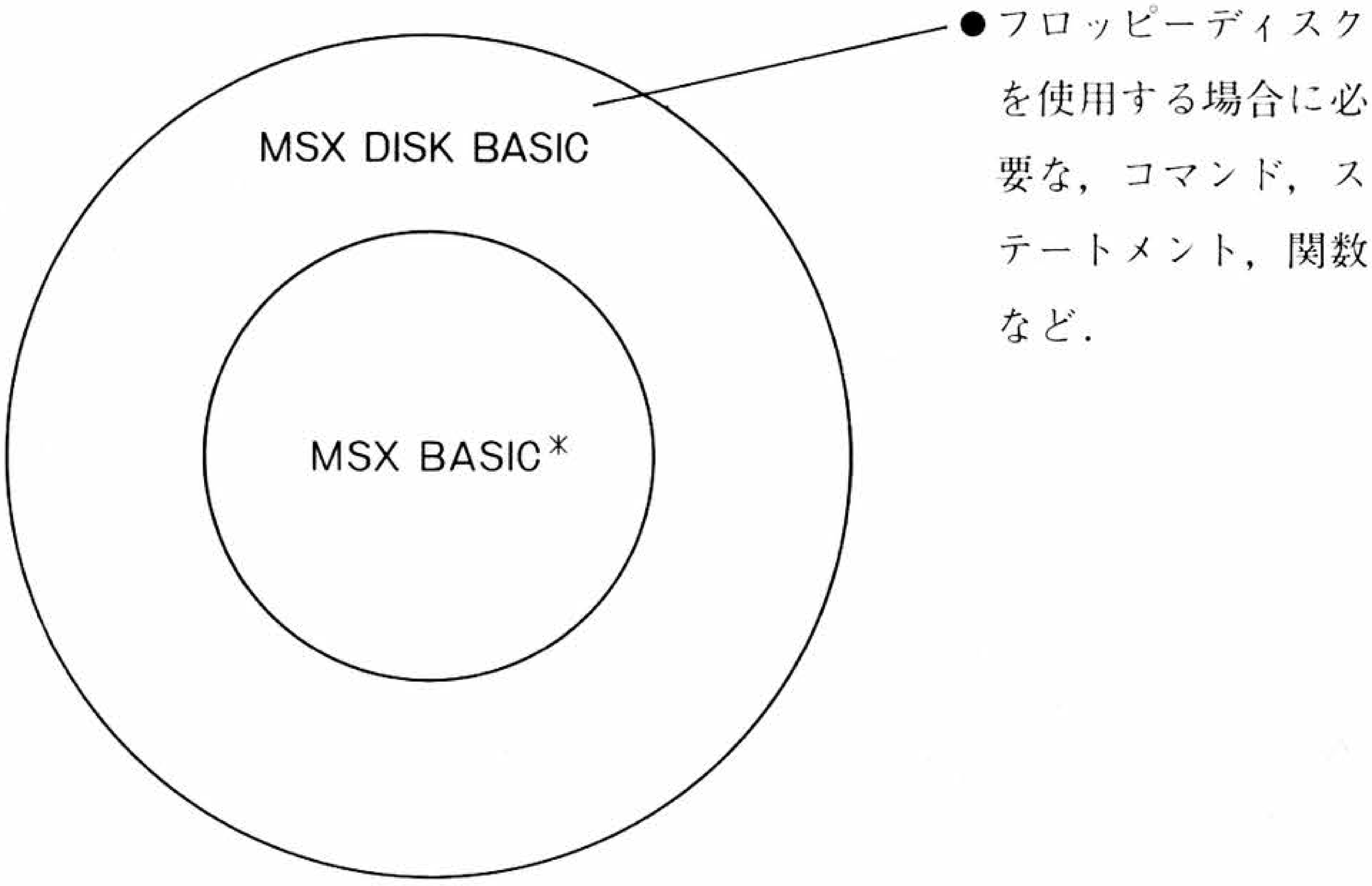
第1章

DISK BASICの概要

1.1 DISK BASICとは

MSX DISK BASICはマイクロソフト社の標準DISK BASICバージョン1.0を拡張したものです.

MSX BASICと，MSX DISK BASICは，次のような関係になっています.



＊MSX BASIC……マイクロソフト社の標準BASICバージョン4.5を拡張したもので，MSXパーソナルコンピュータ用にグラフィック機能および，音楽機能などが追加されています.

DISK BASICという言葉自体はいろいろな意味で使われますが，上図のように，フロッピーディスクをメディアとして使用する（メモリー上のデータをディスク上に記録したり，反対にディスクからメモリーへ呼び出したりする)ための命令や関数を含むBASIC言語全体のことをDISK BASICと呼びます.

このように，MSX DISK BASICとは，MSX BASICを含んだ，フロッピーディスクを使用するための言語です.

プログラムの作成などは，MSX BASIC言語を使用します.
(MSX BASICについては，お手持ちのMSX BASIC説明書などをよくお読みください.)

1.2 フロッピーディスクについて

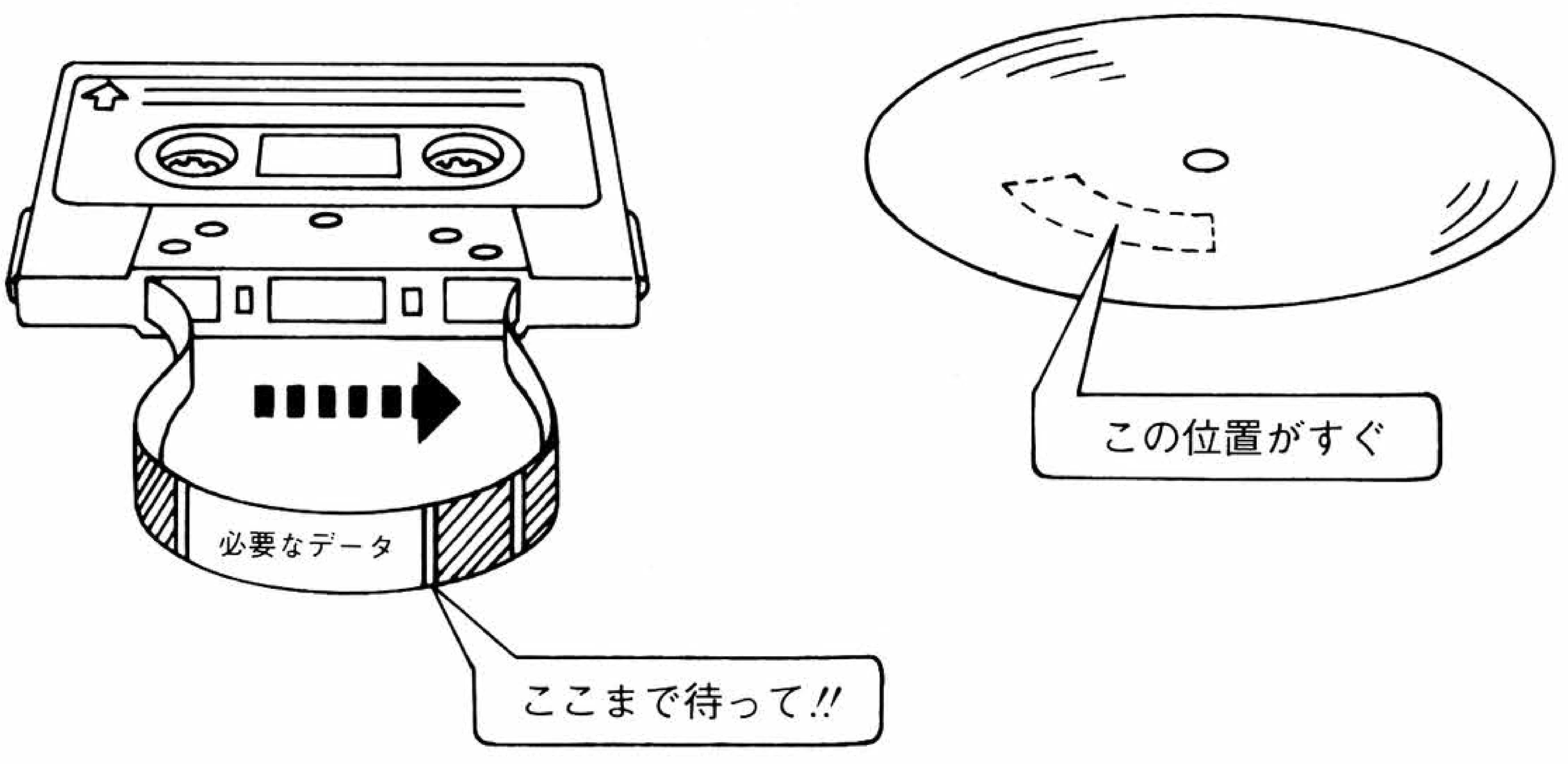
DISK BASICを使用する目的が「データの入出力に使用するメディアにフロッピーディスクを使う」と「1.1 DISK BASICとは」で説明しましたが，ここでは，そのメディアであるフロッピーディスクについて説明します．

1.2.1 フロッピーディスクとカセットテープは…

データの入出力用メディアとして，フロッピーディスクとカセットテープは，どのように違うのかを説明しましょう．

カセットテープは，いつもお使いになる音楽テープなどと同じですが，フロッピーディスクは，レコード盤のようなものと，お考えください．

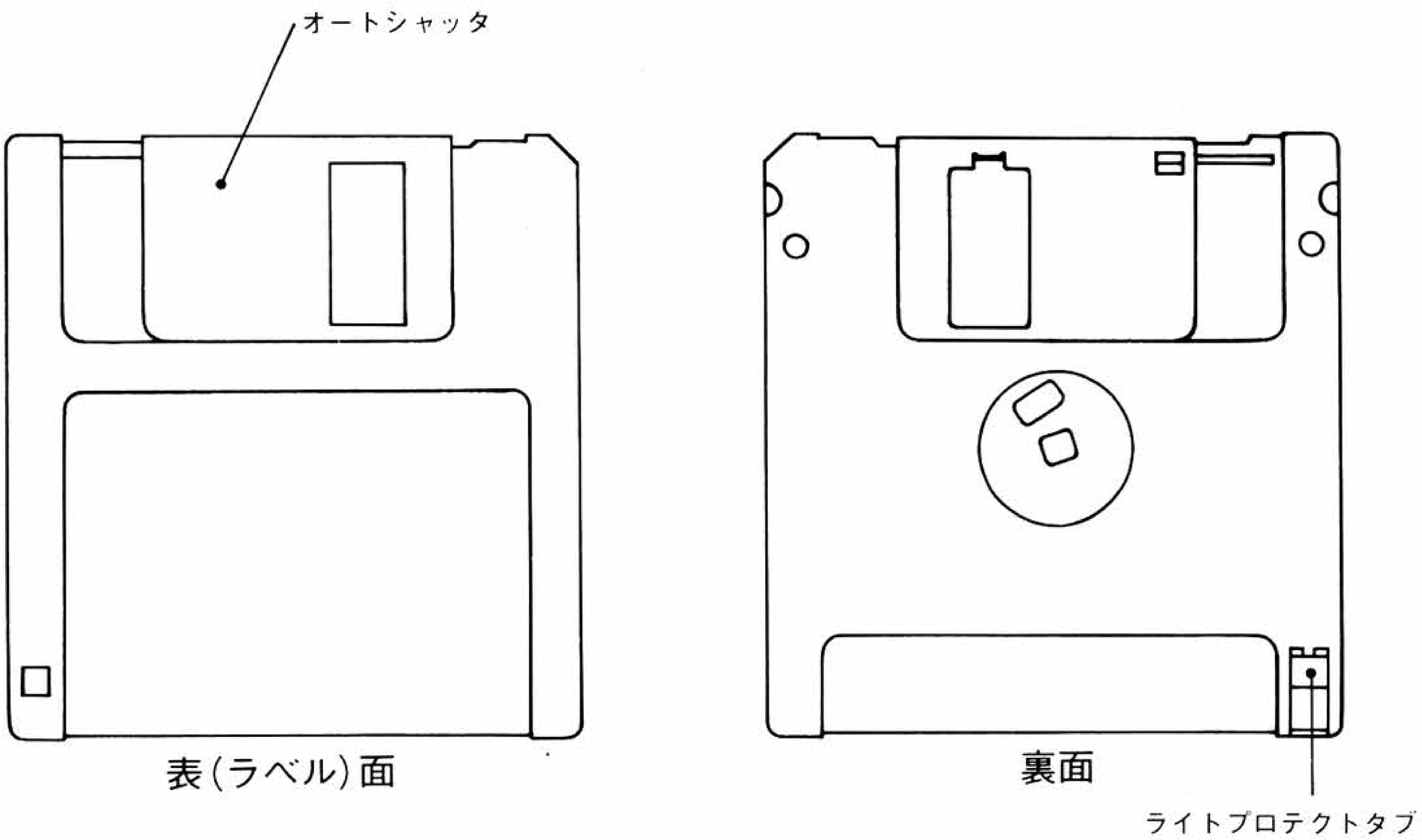
カセットテープは，基本的に1曲目，2曲目……と進んで行きます．好きな曲を聞きたいときなどは，めんどうでも，早送り，巻き戻しなどをおこなう必要がありますが，レコードは，目的の曲に針を降すだけで聞くことができます．



このように，フロッピーディスクはカセットテープよりも，必要なプログラムやデータを素早く，正確に探し出し，また，データなどを全て読み取る時間も，カセットテープよりはるかに速く，読み違いの可能性もずっと少ないのです．

1.2.2　3.5インチマイクロフロッピーディスク

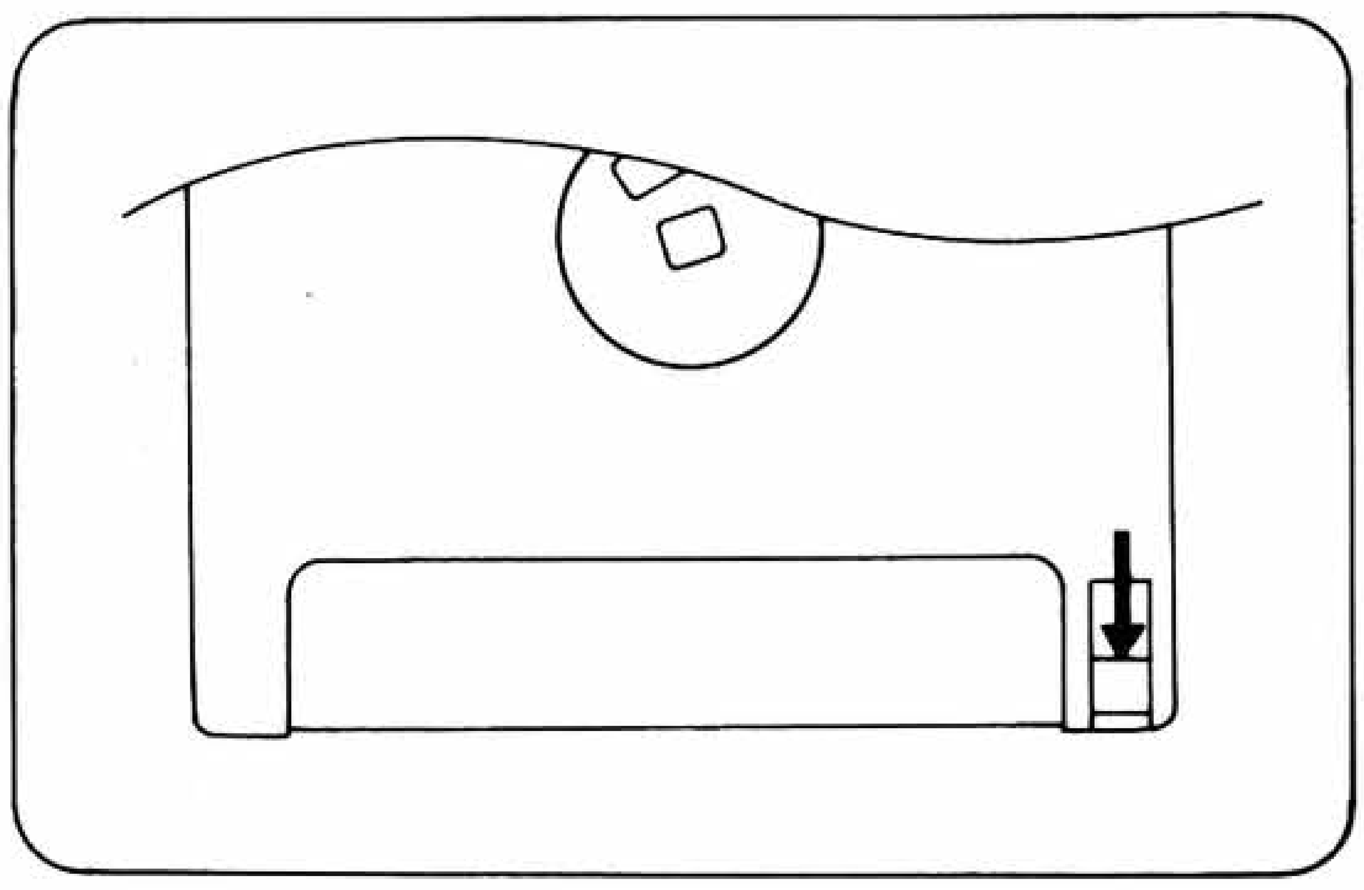
MSX DISK BASICでメディアとして使用するフロッピーディスクは，ほとんどが3.5インチフロッピーディスクで，下図のような外観をしております．



●ライトプロテクトについて

フロッピーディスクの裏面についているライトプロテクトタブは，書き込みを防止する働きをします．

大切なデータやプログラムを保存しているフロッピーディスクは，ライトプロテクトタブをペン先などで矢印で示した方向にスライドさせてください．



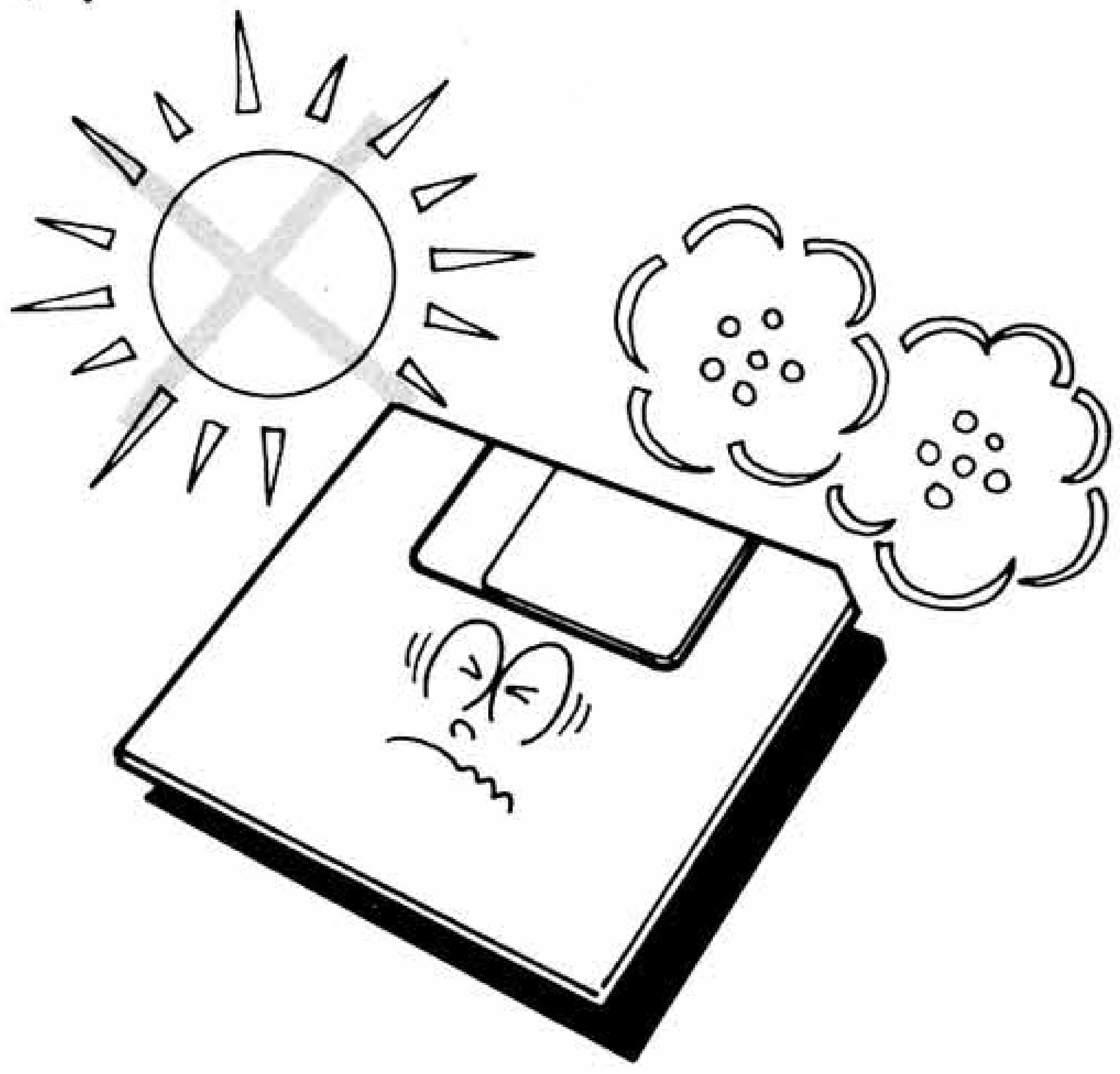
●使用するフロッピーディスクについて

当社製3.5インチマイクロフロッピーディスク(AF-MF2DD11，別売り)をお奨めします．

1.2.3 フロッピーディスクの取り扱いについて

フロッピーディスクは高速でプログラムやデータのセーブ，ロードをおこなうため，ディスク自体の材質およびケースの機構とも，精密に作られています．取り扱われるときは以下の注意事項にしたがって，お使いください．

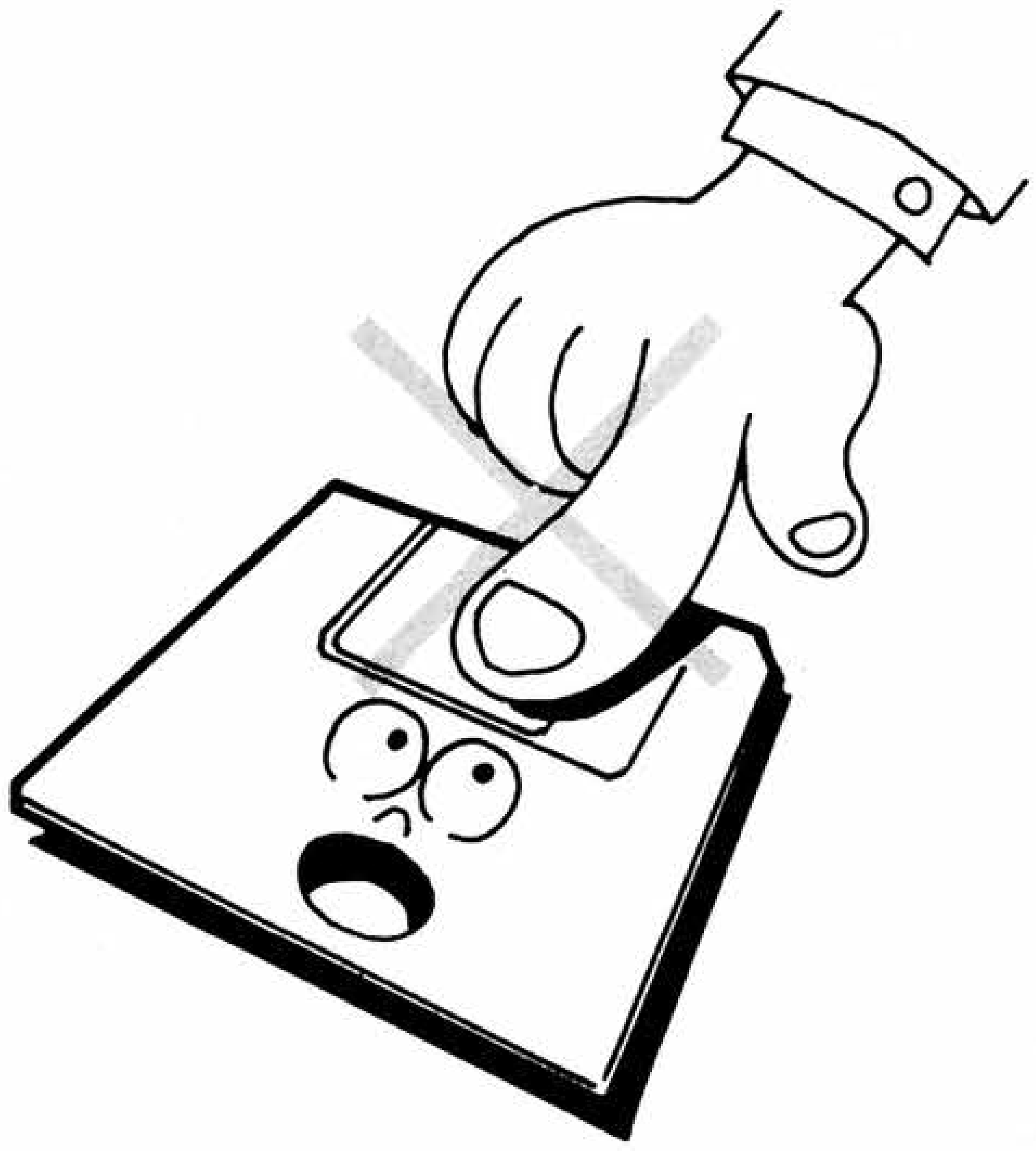
- 高温，低温，直射日光および，ほこりの多い所での使用はさけてください．



- コーヒー，ジュースなどの飲物や花びんの水などをこぼさないでください．
  また，重い物の下に敷かないでください．

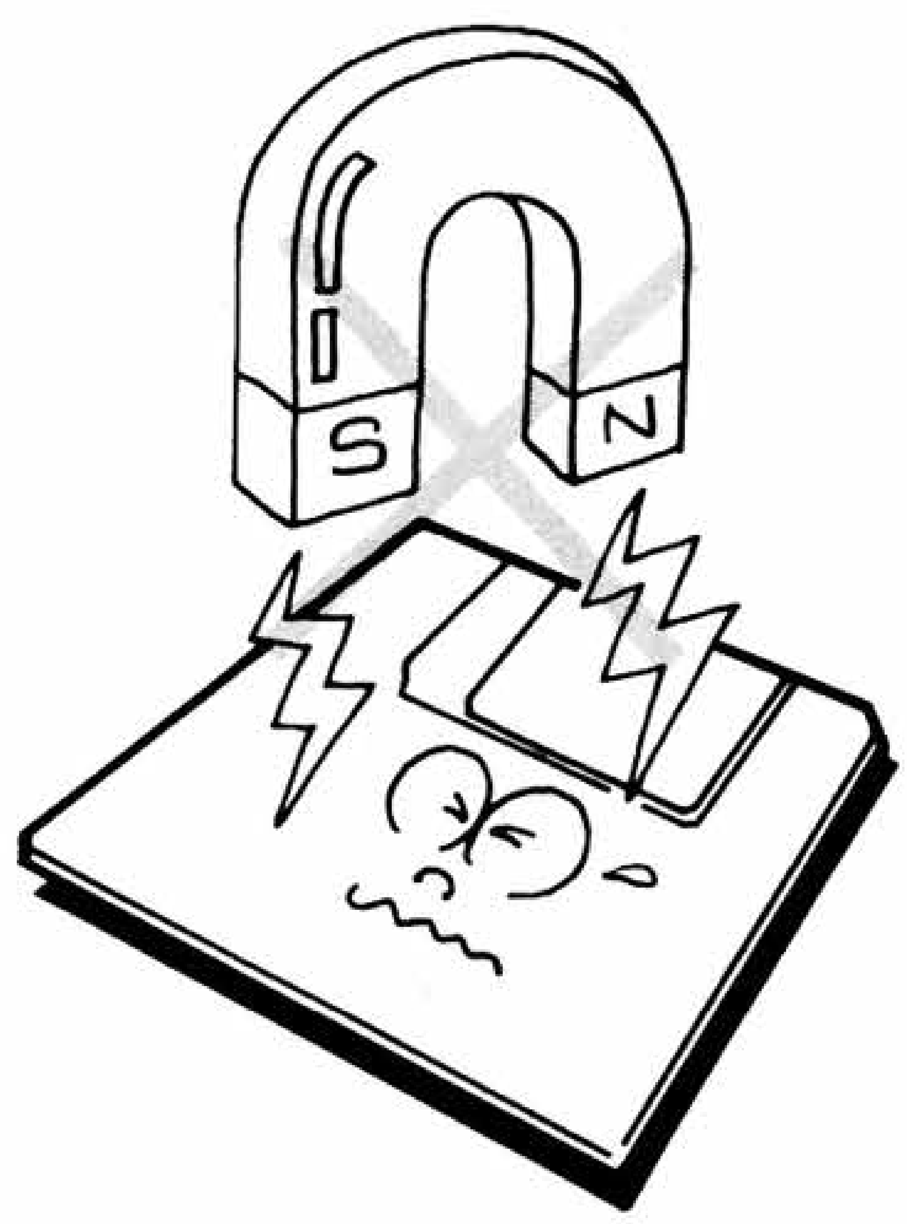


- オートシャッタを指で開けたり，直接ディスク面を指で触れたりしないでください．

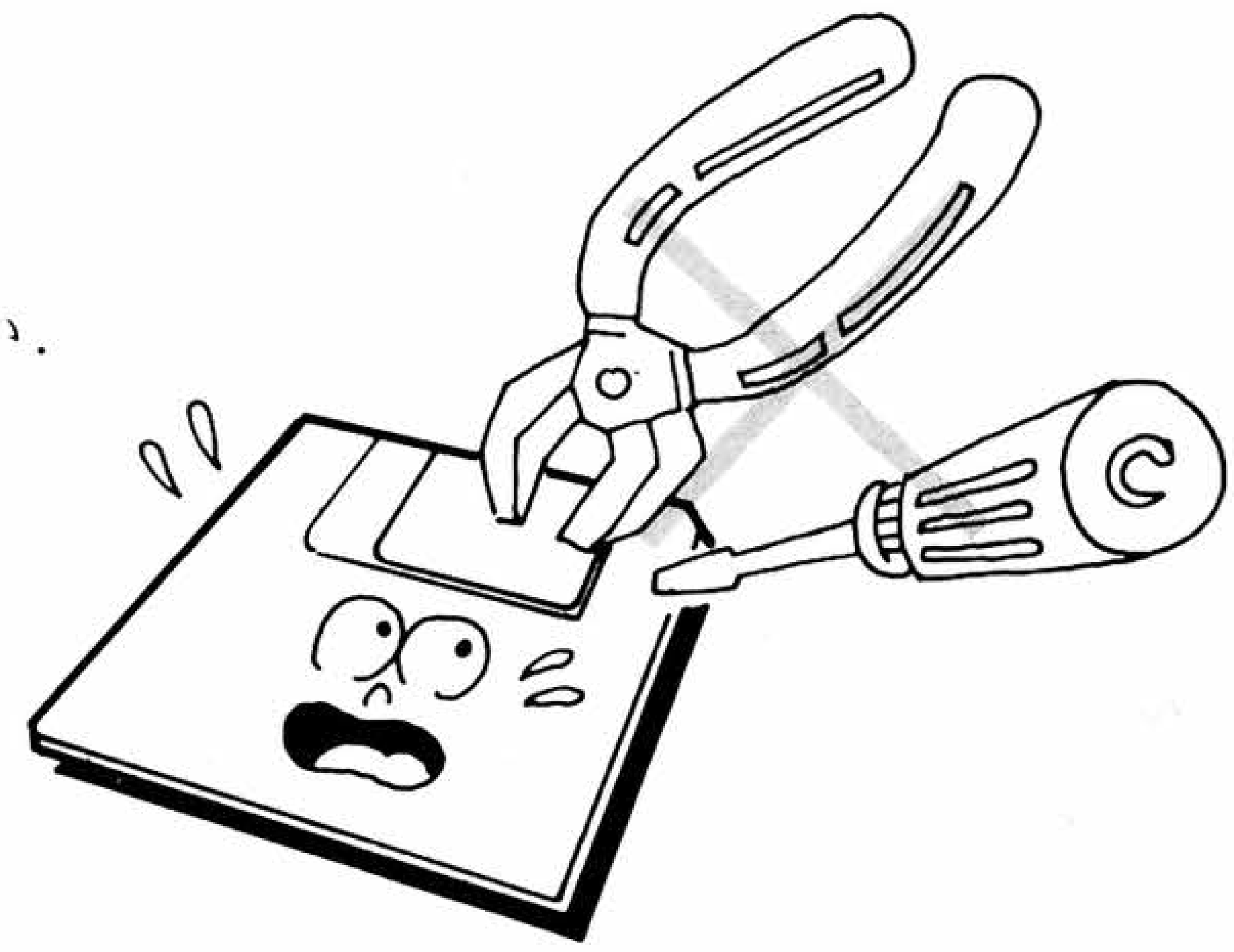


- 寒い場所から急に暖かい場所へ持ち込むと，ディスク表面に水滴が付着(結露状態)し，プログラムやデータが正常にセーブ，ロードできないことがありますので，ご注意ください．

●磁気のある所（スピーカやモータの近く）での
保管はさけてください．
また，磁石は絶対に近づけないでください．



●分解しないでください．
故障の原因になります．
また，異常時は販売店にご相談ください．



■フロッピーディスク使用時のご注意

フロッピーディスクの出し入れ，および操作については，次の注意事項にしたがって注意して，おこなってください．（ディスク上のデータをこわす恐れがあります．）

1．ディスクインジケータが点灯しているとき（セーブまたはロード時）に，フロッピーディスクの出し入れをしないでください．

2．セーブまたはロード時に，電源を切らないでください．

1.3 フォーマットについて

新しく購入された未使用のフロッピーディスクは，使用される前にフォーマットする必要があります．フォーマットしないで使用するとDISK BASICではDisk Offlineのエラーメッセージが表示されます．

（フォーマットについて詳しくは，「1.3.2 フォーマットとは」を参照してください．）

1.3.1 フォーマットのしかた

フォーマットをおこなうと，そのフロッピーディスクにセーブされていた内容は，すべて消去されます．貴重なディスクを誤ってフォーマットしないように注意してください．

フォーマットは，DISK BASICのFORMATコマンドを使用します．

操作は，すべて画面と対話形式でおこないます．

1．コンピュータをDISK BASICが使用できる状態にします．

2．CALL FORMATと入力し，リターンキーを押します．

3．右のようにドライブ名を聞いてきます．

```
Drive name? (A,B)■
```

●使用するディスクドライブ（AまたはB）名を入力してください．

＊通常はAです．（詳しくはご使用機器の取扱説明書をご覧ください．）

4．続いて，フォーマットの種類を聞いてきます．フォーマットの種類は，機種により異なります．（詳しくはご使用機器の取扱説明書をご覧ください．）

```
1 - 1 side
2 - 2 sides
3 - 1 side, double track
4 - 2 sides, double track
?■
```

●使用するフロッピーディスクのタイプに合わせて，1～4を入力してください．

＊フォーマットの種類とフロッピーディスクのタイプ

1 - 1 side……………………………片面倍密度記録（1D）

2 - 2 sides …………………………両面倍密度記録（2D）

3 - 1 side, double track ……片面倍密度倍トラック記録（1DD）

4 - 2 sides, double track……両面倍密度倍トラック記録（2DD）

（当社製MSXパーソナルコンピュータの推奨フロッピーディスク（AF-MF2DD11）の場合は，「4」です．）

5．右のようなメッセージが現れます．

```
Strike a key when ready■
```

●リターンキーを押すとフォーマットが開始されます．

6．右のようなメッセージが現われると，フォーマット完了です．

```
Format complete
Ok
■
```

1.3.2 フォーマットとは

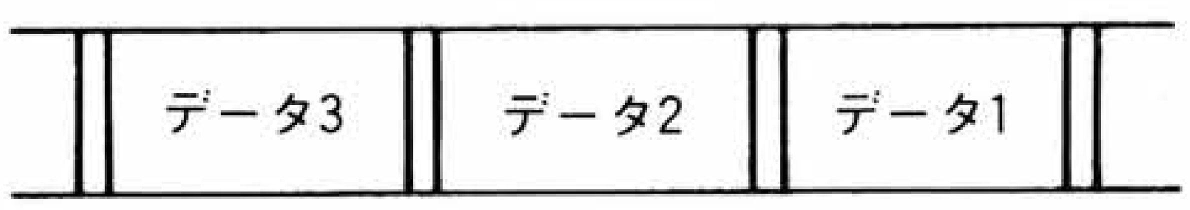
フロッピーディスクは，カセットテープのようにプログラムやデータをセーブされた順に1列に並べて記録するのではなく，フロッピーディスク上をいくつかのエリア（区画）にあらかじめ区切っておき，そのエリア内に記録していきます．

この記録エリアに区切ることを，フォーマットと言います．

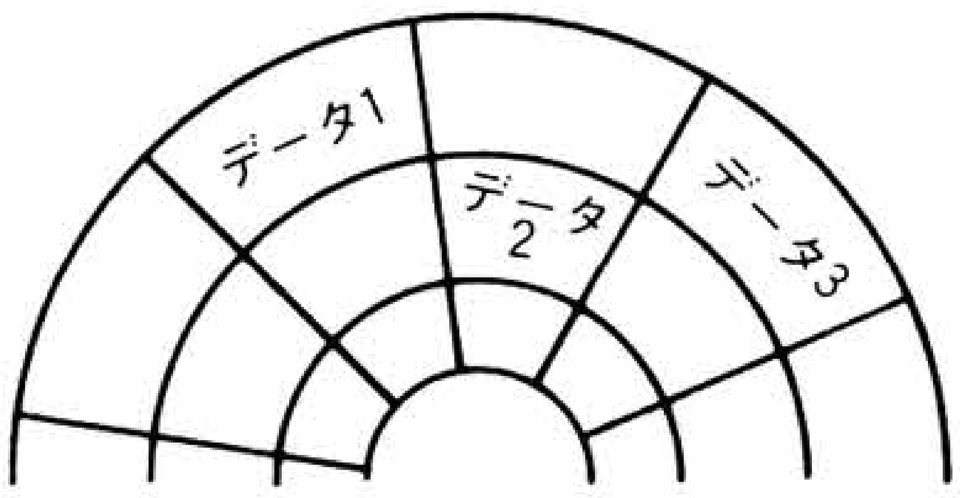
＊フロッピーディスク上の記録エリアのことをセクタと呼びます．

セクタの境界線は，ほとんどの場合，プログラムで書き込まれます．

これをソフトセクタ方式と呼びます．



カセットテープ



フロッピーディスク

参考

フロッピーディスクの中心から同心円上の境界をトラックと呼び，放射状の境界をセクタと呼びます.(14ページ図1参照)

両面倍密度倍トラックの場合，フロッピーディスク上にトラックが両面で160トラック，セクタは1トラック当り9セクタありますので記録エリアは，

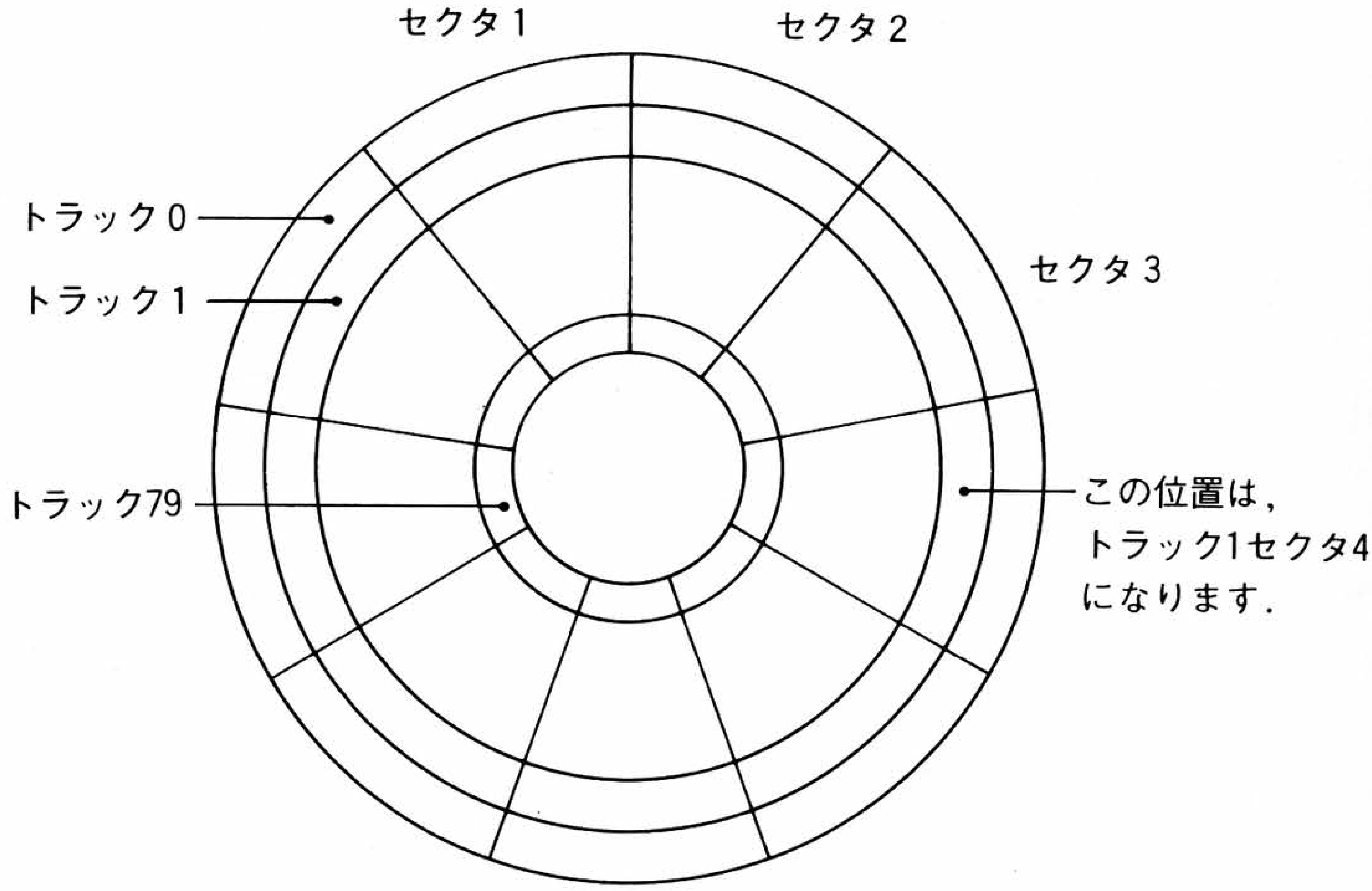
$$160 \times 9 = 1440$$

1440セクタになります．

また1セクタには，512バイトのデータが記録できますので，1枚のフロッピーディスクには，

$$1440 \times 512 = 737280$$

737280バイトのデータが記録できます．



〔図1〕

●物理セクタと論理セクタ

物理セクタとは

フロッピーディスク上の特定領域を表わすときに，「トラック○○のセクタ××」と表わす方法を物理セクタと呼びます．

論理セクタとは

フロッピーディスク上のトラック0，サイド0，セクタ1を「セクタ0」とし，順番に数えていく方法を論理セクタと呼びます．

物理セクタ	トラック0 サイド0 セクタ1～9	トラック0 サイド1 セクタ1～9	トラック1 サイド0 セクタ1～9	………………………………
論理セクタ	セクタ0～8	セクタ9～17	セクタ18～26	………………………………

1.4 ファイルについて

ファイルとは意味を持つ情報の集まりのことです．

フロッピーディスクにしまったプログラムやデータはファイルとして扱われます．

1.4.1 ファイルスペック

ファイルスペックは，ファイルの入出力の際に対象となる入出力機器やファイル名を指定します．ファイルスペックは次のような形式の文字列で表します．

●"〈デバイス名〉：〈ファイル名〉.〈拡張子〉"

ファイルスペックは，必ず引用符（"）で囲みます．

〈デバイス名〉，〈ファイル名〉，〈拡張子〉とも場合によって省略することができます．

1.4.2 デバイス名

〈デバイス名〉は入出力機器を指定するものでフロッピーディスクの場合A～Hです．

通常は，1台目のドライブに対してはAを使用します．2台以上増設した場合は，増設のしかたにより，〈デバイス名〉が異なりますので，ご使用の機器の取扱説明書を参照してください．

〈**例**〉• パソコン本体にドライブを内蔵していない場合の増設（FS-A1 にFS-FD1を接続した場合など）．

		1台増設	2台増設
デバイス名	スロット1側	A	A
	スロット2側		C

〈**参考**〉• コピーを作成するような場合，1台のドライブにAおよびBのドライブ名を割り当てて使用することもできます．詳細は「1.6ディスクコピーのしかた」(18ページ）を参照してください．

• 〈デバイス名〉を省略すると，Aとみなされます．

例）LOAD " A：SAMPLE. BAS " とLOAD " SAMPLE. BAS " は，どちらもドライブAが選択されます．

1.4.3 ファイル名

〈ファイル名〉とはプログラムファイルやデータファイルに付ける名前です．

〈ファイル名〉は8文字までの文字列です．また〈ファイル名〉が8文字を越えて指定された場合先頭から8文字が〈ファイル名〉と見なされます．

●〈ファイル名〉は英数字，カタカナ，ひらがなが使用できます．また英字の大文字と小文字は区別されません．

（カセットテープの場合と異なりますのでご注意ください．）

1.4.4 拡張子

〈拡張子〉とは，ファイルの種類を区別するためにつけておくもので，3文字で付けます．ディスクの中にファイルの数が増えてきたときに大変便利です．

●〈ファイル名〉の後にピリオド(．)を打って〈拡張子〉を入力します．

〔例〕

BASICプログラム ………BAS

機械語プログラム…………BIN

データ……………………DAT　など．

●〈拡張子〉は大文字と小文字の区別はされません．

●〈拡張子〉は省略することはできますが,〈拡張子〉を付けてセーブしたファイルをロードする場合は省略することができません．

1.4.5 ワイルドカードとは

〈ファイル名〉または〈拡張子〉に対して＊(アスタリスク)や？(クェスチョン) を任意の文字列や文字(または数字)として使用することができます．

この＊や？をワイルドカードと呼びます．

例) 次のようなファイルがディスク上にある場合，ワイルドカードを使って以下のように操作できます．

ABC.BAS

ACB.BAS

AB1.BIN

↓

FILES"AB?.＊"　　●ファイル名の3文字目を？(任意の1文字),拡張子を＊(任意の文字列)として呼び出す．

ABC.BAS

AB1.BIN

FILES"＊.BAS"　　●ファイル名を＊(任意の文字列),拡張子をBASとして呼び出す．

ABC.BAS

ACB.BAS

1.5 セーブとロード

フロッピーディスクにプログラムやデータをセーブ，ロードするときは，カセットテープを使用するときとは異り,「1.4 ファイルについて」で説明しましたファイルスペックが必要です.

1.5.1 セーブのしかた

1．コンピュータをDISK BASICで使用する状態にします.

2．フォーマットされたフロッピーディスクをディスクドライブに装着します.

3．プログラムを入力します.

4．プログラムに適当なファイル名を付け,
SAVE"<デバイス名>：<ファイル名>.<拡張子>"
と入力し，リターンキーを押します.

●ディスクドライブインジケータが点灯し,「セーブ」が始まります.
「セーブ」が終了すると画面に「Ok」が表示されます.

〔例〕

SAVE"A：SAMPLE. BAS"

SAVE"A：NAME. BAS"

1.5.2 ロードのしかた

1．コンピュータをDISK BASICで使用する状態にします.

2．必要なプログラムやデータがセーブされているフロッピーディスクを，ディスクドライブに装着します.

3．LOAD"<デバイス名>：<ファイル名>.<拡張子>"
と入力し，リターンキーを押します.
(<ファイル名>,<拡張子>は「セーブ」したときの内容を入力してください.)

●ディスクドライブインジケータが点灯し,「ロード」が始まります.
「ロード」が終了すると画面に「Ok」が表示されます.

〔例〕

LOAD"A：SAMPLE. BAS"

LOAD"A：NAME. BAS"

1.6 ディスクコピーのしかた

フロッピーディスクは，取扱いが便利なうえディスク操作も簡単なために，ちょっとした不注意でディスク上のプログラムやデータをこわしてしまうことがあります．

大切なプログラムやデータは，もう一枚のフロッピーディスクにコピーしておかれることをお奨めします.(このことをバックアップと言います.)

ここでは，一台のディスクドライブでコピーをする方法を説明します．

1．コンピュータをDISK BASICを使用する状態にします．

2．作成したプログラムやデータの入ったフロッピーディスクをディスクドライブに装着します．

3．COPY"A：<ファイル名>.<拡張子>"TO"B："と入力し，リターンキーを押します．

●フロッピーディスクからプログラムやデータをメモリーへロードし，右のようなメッセージが返ってきます．

```
Insert diskette for drive B:
and strike a key when ready■
```

4．ここで，ディスクドライブからプログラムやデータの入ったフロッピーディスクを抜き取り，コピー用の別のフロッピーディスクを装着します．

5．リターンキーを押します．

ディスクドライブインジケータが点灯し，「コピー」が始まります．

●ファイルが小さくてコピーが完了したときは,「Ok」が表示されますが，ファイルが大きいときは，右のようなメッセージが返ってきます．

```
Insert diskette for drive A:
and strike a key when ready■
```

これは，大きなファイルをコピーするときは，ファイルをいくつにも分けてロード，セーブするためで,「次の部分をコピーするので，プログラムやデータの入ったフロッピーディスクをもう一度装着してください.」と言う意味です．

6．コピー用のフロッピーディスクを抜き取り，
　プログラムやデータの入ったフロッピーディスクを装着します．

7．リターンキーを押します．
　●再び，
　Insert diskette for drive B：
　and strike a key when ready■
　と表示されます．

大きなファイルをコピーするときは，4．5．6．7．の操作を「Ok」が表示されるまで，くり返しおこなう必要があります．

通常短かいプログラムでも2～3回，長いプログラムでは数十回の繰り返しが必要です．

長いプログラムをコピーされる場合は，MSX-DOSのCOPYコマンドを使用されると便利です．
（詳しくは，第4章を参照してください．）

ご注意：繰り返し操作をされている時，プログラムやデータの入ったフロッピーディスクとコピー用のフロッピーディスクの装着の順序を間違わないようご注意ください．
また，プログラムやデータの保護のため，重要なプログラムやデータを保存したフロッピーディスクはライトプロテクトしておくことをお奨めします．
（9ページ参照）

1.7 BASICプログラムの自動立ち上げ

DISK BASICは，電源を入れた時またはリセットボタンを押した時，"AUTOEXEC.BAS"というファイル名と拡張子を持ったファイルがあると，このファイルを自動的にロードし実行する機能があります．

これによりBASICプログラムの自動立ち上げが可能です．

フロッピーディスクの操作に慣れていない人に使用していただくと便利な機能です．

●すでに作成されたプログラム（例"SAMPLE.BAS"）を自動立ち上げさせるため"AUTOEXEC.BAS"に変更するには，以下のようにしてください．

```
NAME "SAMPLE.BAS"AS"AUTOEXEC.BAS"
```

ご注意：ご使用のフロッピーディスクに"MSXDOS.SYS"というMSX-DOSのシステムファイルがある場合は，"AUTOEXEC.BAS"での自動立ち上げはできません．

1.8 DISK BASIC(コマンド,ステートメント,関数)の分類と関係

DISK BASICのコマンド，ステートメント，関数には，これまで説明してきた以外にも多くの種類がありますが，すべてを一度に理解される必要はありません．

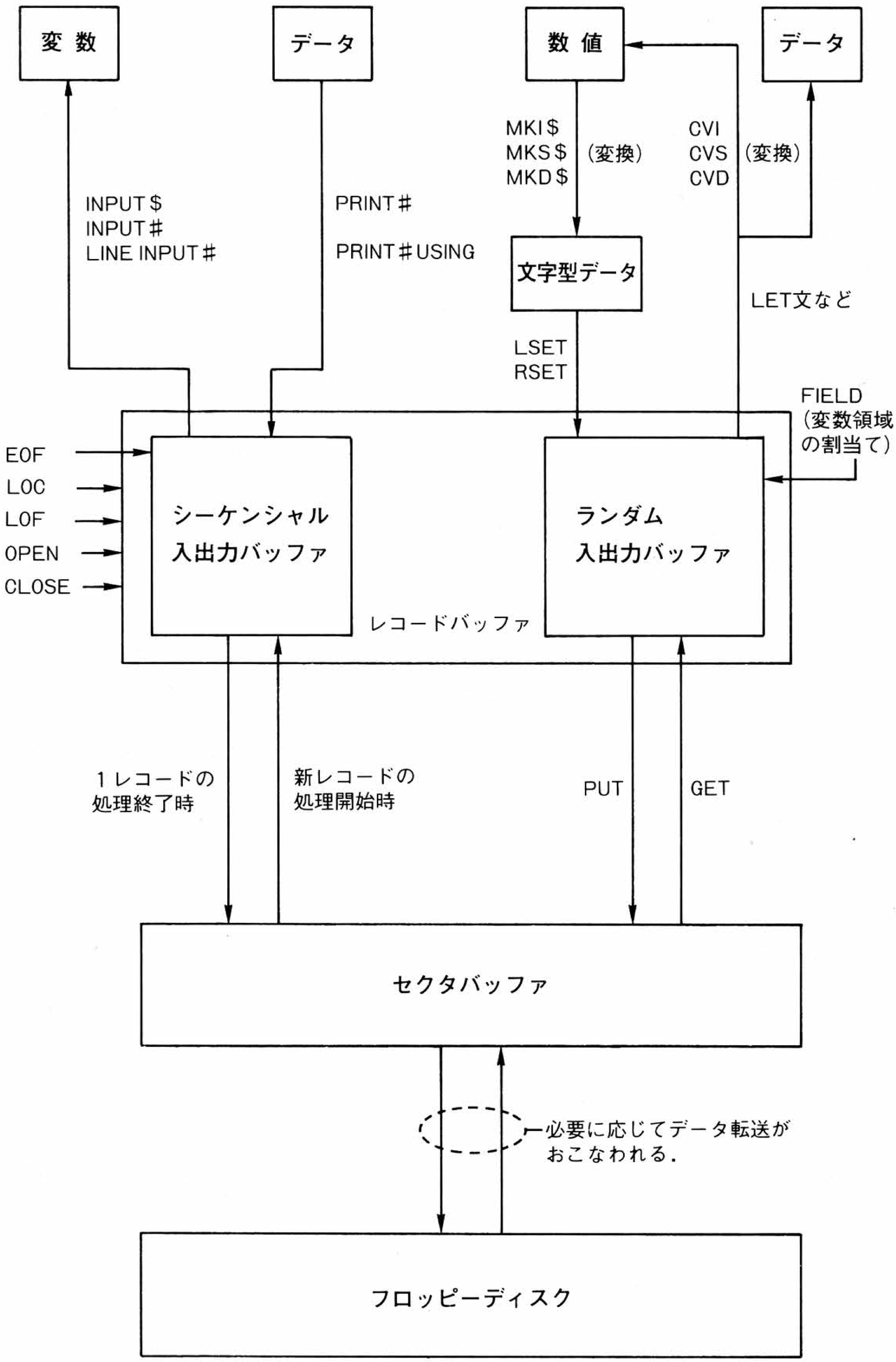
必要に応じて，順次理解していってください．

また，このページにDISK BASIC(コマンド，ステートメント，関数)の分類，21ページには，ファイル操作に関するコマンド，ステートメント，関数の関係を図に示していますので参考にしてください．

■DISK BASIC(コマンド,ステートメント,関数)の分類

分類		コマンド，ステートメント，関数
フロッピーディスクへの書込み，読取り	書込み	SAVE，BSAVE，DSKO$
	読取り	LOAD，BLOAD，DSKI$，RUN
ファイル操作関係	データの転送	INPUT#，INPUT$ LINE INPUT#，PRINT# PRINT# USING，PUT，GET LSET，RSET
	バッファの設定 状態調査	OPEN，CLOSE，FIELD EOF，LOC，LOF
	データの変換	MKI$，MKS$，MKD$ CVI，CVS，CVD
ユーティリティ		FORMAT，COPY，FILES KILL，MERGE，DSKF NAME，SYSTEM

■ファイル操作に関するコマンド,ステートメント,関数の関係



第2章

DISK BASICリファレンス

DISK BASICのコマンド，ステートメント，関数

第2章の見方

初心者の方は，命令名の右上に“*”を付した部分は，読みとばしてください．

“*”の付いていない命令を理解された後で読まれた方が理解が容易です．

● **機　能**　命令の機能を簡単に示します．

● **書　式**　命令の書き方を示します．

実際の入力時には次のような決まりに従ってください．

- ●英字の大文字で示された項目は，そのとおり入力します．入力する場合は，小文字でも大文字でもかまいません．
- ●カギカッコ〈　〉で囲まれた項目は，ユーザーが指定します．
- ●角カッコ〔　〕で囲まれた項目は，オプションであり，省略することができます．省略した場合，省略値（BASICによって設定される値）または以前に指定した値が適用されます．
- ●下記のようにパラメータの後にセミコロン（；）を指定するような場合は，セミコロンも含めて省略します．

PRINT#〈ファイル番号〉，〔USING〈書式〉;〕〔〈式〉〔;〈式〉…〕

BLOAD

ビーロード(コマンド)

● **機　能**　**機械語プログラム，ビデオRAMの内容をロードする．**

● **書　式**　**BLOAD"〈ファイルスペック〉"{〔,R〕 | 〔,S〕}〔,オフセット〕**

● **文　例**

BLOAD"A：DEMO",R
BLOAD"A：TEST",&H100

● **解　説**　〈ファイルスペック〉で指定した機械語プログラムファイルをメモリ上にロードします．ファイルがカセットテープ上にある場合はファイル名を省略できますが，ディスク上にある場合はファイル名の省略はできません．

Rオプションを指定するとプログラムロード後，直ちに実行されます．このとき既に開かれているファイルがあればファイルは開かれたままプログラムを実行します．

Sオプションを指定すると，ビデオRAMにロードします．

〈オフセット〉を省略すると，機械語プログラムはBSAVEでセーブするときに指定した〈開始アドレス〉からロードされます．

〈オフセット〉を指定した場合には，セーブの際に指定した開始アドレスに〈オフセット〉を加えた番地からロードします．したがって〈オフセット〉を付けてロードするプログラムは，セーブされたときと異なる番地にロードされるので，リロケータブル(再配置可能)なプログラムでなければなりません．

〈ファイルスペック〉については「1.4ファイルスペックについて」を参照してください．

● **参　照**　BSAVE

サンプル
プログラム

★このプログラムを入力する前に「BSAVE」27ページの例1のプログラムを入力してください.
(画面の上，下が黒色になります.)

```
BLOAD"SAMPLE",R
```

★このプログラムを入力する前に「BSAVE」27ページの例2のプログラムを入力してください.
(自動車の絵が現れます.)

```
10 SCREEN 2
20 BLOAD"CAR",S
30 GOTO 30
```

BSAVE

ビーセーブ(コマンド)

● **機　能**　**機械語プログラム，ビデオRAMの内容をセーブする．**

● **書　式**　**BSAVE "〈ファイルスペック〉",〈開始アドレス〉,〈終了アドレス〉{〔,〈実行開始アドレス〉〕 | 〔,S〕}**

● **文　例**　**BSAVE "A：DEMO",&HA700,&HA2FF**

● **解　説**　〈開始アドレス〉から〈終了アドレス〉までのメモリ上の機械語プログラムを〈ファイルスペック〉で指定したファイルにセーブします．
〈実行開始アドレス〉を省略すると〈開始アドレス〉が〈実行開始アドレス〉として登録されます．
Sオプションを指定するとビデオRAMの内容をファイルにセーブします．
MSX2 BASICの場合，指定するビデオRAMのアドレスは，SET PAGE文で設定したページの先頭からの〈オフセット〉で0～65535の値です．
ドライブ名を省略すると，現在使用しているドライブにセーブします。

● **参　照**　**BLOAD**

**サンプル
プログラム**

例1）

```
100 'BSAVE ノ サンプ゜ル プ゜ログ゛ラム
110 'キカイゴ" ノ セット
120 FOR AD=&HD000 TO &HD008
130 READ D$:D=VAL("&H"+D$)
140 POKE AD,D
150 NEXT AD
160 DATA 3E,01,32,EB,F3,CD,62,00,C9
170 '
180 BSAVE"SAMPLE",&HD000,&HD008
```

例2）

```
100 'ガ"メン ノ SAVE
110 SCREEN 2
120 LINE(80,90)-(170,120),,BF
130 LINE(110,90)-(160,70),,BF
140 CIRCLE(150,120),13,1:PAINT(150,120),1
150 CIRCLE(100,120),13,1:PAINT(100,120),1
160 BSAVE"CAR",0,&H37FF,S
```

CLOSE

クローズ(ステートメント)

● **機　能**　ファイルを閉じる.

● **書　式**　**CLOSE〔〔#〕<ファイル番号>〔,〔#〕<ファイル番号>〕…〕**

● **文　例**　**CLOSE　#1**

● **解　説**　<ファイル番号>に対応するファイルを閉じます.

CLOSE文は, ファイルが出力用にオープンされている場合, バッファ上に残っているデータをすべて掃き出しますので, ファイルへの出力処理を正しく終了するためには, 必ずCLOSE文を実行しなければなりません.

CLOSE文で指定した<ファイル番号>は以後のOPEN文で自由に指定することができるようになります.

閉じたファイルに対しては再びOPEN文で開くまで入出力を行うことはできません.

END文, CLEAR文, MAXFILES文, NEW文を実行すると、自動的にすべてのファイルを閉じますが, STOP文はファイルを閉じませんので注意してください.

● **参　照**　OPEN

COPY

コピー(コマンド)

● **機　能**　ファイルをコピーする.

● **書　式**　**COPY"<ファイルスペック>"TO"<ファイルスペック>"**

● **文　例**　**COPY"A：DEMO" TO"B：DEMO 2"**

● **解　説**　始めに書かれている<ファイルスペック>で指定されるドライブのファイルを2番目に書かれている<ファイルスペック>で指定されるドライブのファイルにコピーします.

(例) 2番目のファイルスペックでドライブ名だけを指定した場合.

COPY"A：DEMO"TO"B："

始めに書かれている<ファイルスペック>のファイル名と同じファイル名で2番目の<ファイルスペック>で指定されるドライブにコピーされます.

(例) 2番目のファイルスペックでファイル名だけを指定した場合.

COPY"A：DEMO"TO"DEMO2"

2番目の<ファイルスペック>でドライブ名を省略すると自動的にデバイス名がA：となるため,ドライブAを指定したことになります。

このため始めに書かれている<ファイルスペック>で示されるドライブ(この場合ドライブA)に2番目の<ファイルスペック>で指定されるファイル名でコピーされます.

－シングルドライブシステムをマルチドライブシステム仕様で使用する場合－

●「1.6 ディスクコピーのしかた」を参照してください.

CVI, CVS, CVD

シーヴィアイ，シーヴィエス，シーヴィディー(関数)

● **機　能**　**文字列を数値データに変換する．**

● **書　式**　**CVI (<2バイト文字列>)**
CVS (<4バイト文字列>)
CVD (<8バイト文字列>)

● **文　例**　**Y=CVS (N$)**

● **解　説**　ランダムファイルは文字型データのみを扱います．
したがって数値型データは文字型データに変換して書き込み，読み込み時は，逆に文字型データを数値に変換します．
この場合の文字型データを数値型データに変換するのがCVI, CVS, CVDです．
CVIは2バイトの文字列を整数値に
CVSは4バイトの文字列を単精度実数値に
CVDは8バイトの文字列を倍精度実数値に
それぞれ変換します．

● **参　照**　**MKI$，MKS$，MKD$**

**サンプル
プログラム**

```
100 'ﾃﾞｰﾀ ﾉ ｻｸｾｲ
110 OPEN "DATA1" AS #1
120 FIELD #1,4 AS A$
130 FOR I=1 TO 2
140 B$=STR$(I):LSET A$=B$
150 PUT #1
160 NEXT I:CLOSE
170 'ﾗﾝﾀﾞﾑﾌｧｲﾙ ｶﾗ ﾉ ﾆｭｳﾘｮｸ
180 OPEN "DATA1" AS #1
190 FIELD #1,4 AS A$
200 FOR I=1 TO 2
210 GET #1,I:A=CVS(A$):PRINT A
220 NEXT I:END

run
 3.1202E-33
 3.2202E-33
Ok
```

DSKF

ディスクエフ(関数)

● **機　能**　ディスクの残り容量を調べる.

● **書　式**　**DSKF(<ドライブ番号>)**

● **文　例**　**PRINT DSKF(1)**

● **解　説**　<ドライブ番号>で指定されたディスクの残り容量を調べます. 残り容量は1024バイト単位で示されます.

<ドライブ番号>　0と1はドライブA

2はB

3はC

⋮

8はHを示します.

サンプルプログラム

```
100 INPUT "ト゛ライフ゛ ナンハ゛ー";DR
110 PRINT "フリー スヘ゜ース゛="DSKF(DR)*1024"bytes"

run
 ト゛ライフ゛ ナンハ゛ー? 1
 フリー スヘ゜ース゛= XXXXX bytes
Ok
```

DSKI$*

ディスクアイダラー(関数)

● 機　能　ディスクの内容をメモリにロードする.

● 書　式　**DSKI$**(<ドライブ番号>,<論理セクタ番号>)

● 文　例　**A$＝DISK$(1,29)**

● 解　説　&HF351, &HF352番地の内容が示すメモリへ<ドライブ番号> で示すドライブの<論理セクタ番号>で示されるセクタのデータ(512バイト) をロードします.

<ドライブ番号> 0, 1はドライブA, 2はB, 3はC…8はHを示します.

<論理セクタ番号>は0から始まります.

(セクタ番号のチェックはしていませんので注意してください.)

&HF351, &HF352番地の内容はDisk BASICの命令(例えばFILES, OPEN, CLOSE, PRINT#, その他)で破壊されます.

ご注意：メモリーにはユーザーが自由に使えるエリアとシステムが動作上必要なエリアがあり, システムで必要なエリアを誤ってこわすとシステムは正常に動作しないばかりでなく, ディスクのデータをこわしてしまうこともあるのでご注意ください.

● 参　照　DSKO$

DSKO$*

ディスクオーダラー(ステートメント)

● 機　能　メモリの内容をディスクにセーブする.

● 書　式　**DSKO$<ドライブ番号>,<論理セクタ番号>**

● 文　例　**DSKO$ 1, 6**

● 解　説　&HF351, &HF352番地の内容が示すメモリ番地から1セクタ(512バイト)を<ドライブ番号>で示すドライブの<論理セクタ番号>で示されるセクタにセーブします.
<ドライブ番号>0または1はドライブA, 2はB, 3はC… 8はHを示します.
<論理セクタ番号>は0から始まります.
(セクタ番号のチェックはしていませんので注意してください.)
&HF351, &HF352番地の内容はDISK BASICの命令(例えば, FILES, OPEN, CLOSE, PRINT#, その他) でこわされます.

ご注意：ディスクの最初の方のセクタにはディレクトリ, FAT等の重要なディスク情報があります. この部分を誤ってこわすと, そのディスクは使えなくなりますので, 特に注意してください.

ライトプロテクト状態のフロッピーディスクに対して、このステートメントを実行するとエラーメッセージが出ずに“OK”と表示されますが、フロッピーディスクには何も書き込まれません。
実行する前は、メディアのライトプロテクトタブをライト可能な状態にしてご使用ください。

● 参　照　DSKI$

EOF

エンドオブファイル(関数)

● **機　能**　ファイルの終了を調べる.

● **書　式**　**EOF(〈ファイル番号〉)**

● **文　例**　**IF EOF(1)　THEN CLOSE 1**

● **解　説**　〈ファイル番号〉で指定したファイルが終了したかどうかを調べます. もし終了していれば真(−1), 終了していなければ偽(0)の値を返します.
〈ファイル番号〉で指定されるファイルはOPEN文でINPUTモードに開かれていなければなりません.

サンプルプログラム

```
100 'ﾃﾞｰﾀ ﾉ ｻｸｾｲ
110 OPEN "DATA2" FOR OUTPUT AS #1
120 FOR I=1 TO 20
130 PRINT I;:PRINT #1,I
140 NEXT I
150 CLOSE
160 'ﾌｧｲﾙ ﾉ ﾖﾋﾞﾀﾞｼ
170 OPEN "DATA2" FOR INPUT AS #1
180 INPUT #1,I
190 PRINT I;
200 IF EOF(1) THEN END
210 GOTO 180
220 CLOSE

run
 1  2  3  4  5  6  7  8  9
 10  11  12  13  14  15  16
 17  18  19  20  1  2  3  4
 5  6  7  8  9  10  11  12
 13  14  15  16  17  18  19
 20
Ok
```

FIELD

フィールド(ステートメント)

● **機　能**　**ランダム入出力用バッファ中に，変数領域を割り当てる.**

● **書　式**　**FIELD〔#〕<ファイル番号>,<フィールド幅>AS<文字変数>·**
〔,<フィールド幅>AS<文字変数>〕…

● **文　例**　**FIELD #1,15　AS　A$, 10　AS　B$**

● **解　説**　GETやPUT文を実行する前にFIELD文でランダム入出力用バッファの割り当てをしておかねばなりません.

<ファイル番号> はOPEN文でオープンしたファイルの番号です.

<フィールド幅> は <文字変数> に割り当てられる文字数です.

<文字変数> はランダムファイルに入出力するときに使われる変数名です.

FIELD文で割り当てられる総バイト数はOPEN時に決めたレコード長を越えることはできません．もし越えた場合は"Field overflow" エラーが発生します．初期設定値は256バイトです.

同一のファイルに対してFIELD文は何回実行してもかまいません．このとき，以前に実行されたFIELD文も，その効力を保持します.

ご注意：FIELD文で使用した変数名をINPUT文，LET文，その他で使用してはいけません．FIELD文で使用した変数はランダム入出力用バッファ中の対応するデータを示していますが，INPUT 文やLET文等で使用すると，その変数は文字列の記憶領域に移り，ランダム入出力用バッファから消滅してしまいます.

● **参　照**　GET，PUT

サンプル プログラム

```
100 OPEN "TEST.DAT" AS #1
110 FIELD #1,30 AS A$,30 AS B$,30 AS C$
120 'DUMMY$ ﾄﾊ A$+B$+C$ ﾉ ｺﾄﾃﾞｽ
130 FIELD #1,90 AS DUMMY$,2 AS D$,4 AS E$,8 AS F$
140 LSET A$="National"
150 LSET B$="Personal"
160 LSET C$="Computer"
170 RSET D$=MKI$(123)
180 RSET E$=MKS$(12345!)
190 RSET F$=MKD$(.987654321#)
200 PUT #1,1
210 GET #1,1
220 PRINT A$
230 PRINT B$
240 PRINT C$
250 D=CVI(D$)
260 E=CVS(E$)
270 F=CVD(F$)
280 PRINT D:PRINT E:PRINT F
290 END

run
National

Personal

Computer

 123
 12345
  987654321
Ok
```

FILES

ファイルズ(コマンド)

● 機　能　ディスク上のファイル名を表示します.

● 書　式　FILES〔"<ファイルスペック>"〕

● 文　例　FILES "A:"

● 解　説　ファイルスペックで指定されるディスクファイルのファイル名を表示します. ファイルスペックを省略すると, 現在使用しているドライブのファイル名をすべて表示します.ファイル名が見つからないときは"File not found"エラーになり, フォーマットされていないディスクが挿入されたり, ディスクが挿入されていないときは"Disk offline"エラーになります.
（?は1 文字相当, *は文字列相当のワイルドカードの使用ができます.）

例）FILES

```
ABC.BAS
ACB.BAS
AB1.BIN
```

このようなファイル名で各プログラムがセーブされているときに, ワイルドカード(?や*)を使って, ファイル名を表示させると,

```
FILES "AB?.*"
ABC.BAS
AB1.BIN
```

●ファイル名の3文字目を?(任意の1文字), 拡張子を*(任意の文字列)として呼び出す.

```
FILES "*.BAS"
ABC.BAS
ACB.BAS
```

●ファイル名を*(任意の文字列),拡張子をBASとして呼び出す.

★ワイルドカードについての説明は,「1.4.5 ワイルドカードとは」16ページをお読みください.

● 参　照　LFILES

FORMAT

フォーマット(コマンド)

● **機　能**　ディスクの初期化

● **書　式**　{ **CALL FORMAT** / **_FORMAT** }

● **文　例**　**CALL FORMAT**

● **解　説**　CALL FORMATまたは_FORMATと入力すると,あとは画面にメニューが表示されますので,それに従って注意深く入力してください.

●「1.3.1 フォーマットのしかた」を参照してください.

ご注意：フォーマットが終了すると,ディスクは初期化され,すべてのファイルは消滅するので注意してください.

新しいディスクはフォーマットしなければ,使用することはできません.

GET

ゲット(ステートメント)

● **機　能**　**ランダム入出力ファイルから1レコードをランダム入出力バッファへ読み込む.**

● **書　式**　**GET〔#〕<ファイル番号>〔,<レコード番号>〕**

● **文　例**　**GET #2,3**

● **解　説**　<ファイル番号> はOPEN文でランダム入出力ファイルモードで開かれたファイルの番号です.
<レコード番号>を省略すると，最後に実行したGET文，PUT文で指定したレコードの次のレコード番号が割り当てられます.

● **参　照**　**PUT，FIELD**

ご注意：GET文実行後にINPUT#文やLINE INPUT#文を実行すると，ランダム入出力バッファから文字を読んできます.

サンプルプログラム

〔例1〕

```
100 'ﾃﾞｰﾀ ﾉ ｶｷｺﾐ
110 OPEN "DATA3" AS #1
120 FIELD #1,4 AS A$
130 FOR I=1 TO 5
140 B$=STR$(I):LSET A$=B$
150 PRINT I:PUT #1
160 NEXT I:CLOSE
170 'ﾃﾞｰﾀ ﾉ ﾖﾐｺﾐ
180 OPEN "DATA3" AS #1
190 FIELD #1,2 AS A$
200 GET #1:PRINT A$;
210 GOTO 190

run
 1
 2
 3
 4
 5
 1 2 3 4 5
Input past end in 200
Ok
```

〔例2〕

```
100 'ﾃﾞｰﾀ ﾉ ｶｷｺﾐ
110 OPEN "DATA3" AS #1
120 FIELD #1,4 AS A$
130 FOR I=1 TO 5
140 B$=STR$(I):LSET A$=B$
150 PRINT I:PUT #1
160 NEXT I:CLOSE
170 'ﾃﾞｰﾀ ﾉ ﾖﾐｺﾐ
180 OPEN "DATA3" AS #1
190 FIELD #1,2 AS A$
200 FOR I=1 TO LOF(1)/255
210 GET #1:PRINT A$;
220 NEXT I

run
 1
 2
 3
 4
 5
 1 2 3 4 5
Ok
```

★〔例2〕の200行はLOF関数を使って,〔例1〕のように実行時'Input past end…' のエラーメッセージを出さないようにするためのものです.

INPUT ＃

インプットシャープ(ステートメント)

● **機　能**　シーケンシャル入出力ファイルからデータを読み，対応する変数に割り当てる.

● **書　式**　**INPUT＃〈ファイル番号〉,〈変数〉〔,〈変数〉…〕**

● **文　例**　INPUT＃**1**, **A**, **B$**

● **解　説**　あらかじめOPEN文でINPUTモードを指定して開いておいたファイルからデータを読み込み，変数にセットします．INPUT＃文はデータを読み込む対象がファイルであることを除けば，INPUT文とほぼ同じです.

〈ファイル番号〉は，OPEN文で指定した番号です.

INPUT＃文で読み込むデータはPRINT＃文で書き出した各データと対応しています.

〈変数〉の型はファイルから入力するデータの型と対応しなくてはなりません.

サンプルプログラム

```
100 'ﾃﾞｰﾀ ﾉ ｻｸｾｲ
110 OPEN "DATA4" FOR OUTPUT AS #1
120 FOR I=1 TO 5
130 PRINT I;:PRINT #1,I:NEXT I
140 CLOSE
150 'ﾃﾞｰﾀ ﾉ ﾖﾐｺﾐ
160 OPEN "DATA4" FOR INPUT AS #1
170 IF EOF(1) THEN END
180 INPUT #1,A$:PRINT A$
190 GOTO 170:CLOSE

run
 1  2  3  4  5 1
2
3
4
5
Ok
```

INPUT$

インプットダラー(関数)

● **機　能**　**指定されたファイルから指定された長さの文字を読み込む.**

● **書　式**　**INPUT$(〈文字数〉〔,〔#〕〈ファイル番号〉〕)**

● **文　例**　**W$＝INPUT$(4,＃2)**

● **解　説**　〈ファイル番号〉で指定されたファイルから〈文字数〉で指定された文字数のデータを読み込みます.〈ファイル番号〉が省略された場合はキーボードから入力されます. このとき入力した文字は画面には表示されません.
INPUT$文はCTRL+STOP, CTRL+Cを除くすべての文字をそのまま読み込むのでINPUT文やLINE INPUT文では入力することのできないラインフィードコードやキャリッジリターンコードを入力することができます.

● **参　照**　**INPUT, LINE INPUT**

**サンプル
プログラム**

```
100 CLS
110 OPEN "PASS" FOR OUTPUT AS #1
120 I$="NATIONAL"
130 PRINT #1,I$
140 CLOSE
150 OPEN "PASS" FOR INPUT AS #1
160 Y$=INPUT$(8,#1)
170 INPUT "Pass word...";E$
180 IF E$<>Y$ THEN 190 ELSE 210
190 PRINT "You are not user!"
200 GOTO 170
210 BEEP:CLS
220 PRINT "* Welcome to the MSX world *"
230 CLOSE
```

★このプログラムを実行すると, 画面にPass word…?と表示されます.
パスワードは「NATIONAL」ですが, それ以外の文字例を入力すると正しく入力されるまで, パスワードを聞き続けます.

KILL

キル(コマンド)

● 機　能　ディスク上のファイルの消去

● 書　式　KILL "<ファイルスペック>"

● 文　例　KILL "A：DEMO"

● 解　説　<ファイルスペック>で示されるディスクファイルを消去します．
もし，既に開かれているファイルに対してKILL文を実行すると，"File Still open"エラーが発生します．
KILL文はプログラムファイル，ランダム入出力ファイル，シーケンシャル入出力ファイルすべてに有効です．
ワイルドカード(「1.4.5 ワイルドカードとは」16ページ)も使用できますが，使用されるときは十分ご注意ください．

LFILES

リストファイルズ(コマンド)

● 機　能　ディスク上のファイル名をプリンタに印刷します。

● 書　式　LFILES〔"<ファイルスペック>"〕

● 文　例　LFILES "A："

● 解　説　ファイルスペックで指定されるディスクファイルのファイル名をプリンタに印刷します。
LFILESは出力されるものがディスプレイかプリンタかが異なるだけで，その他の機能はFILESと同じです．詳細はFILES文を参照してください．

● 参　照　FILES

LINE INPUT#

ラインインプットシャープ (ステートメント)

● **機　能**　**シーケンシャル入出力ファイルから一行単位のデータを読み、対応する文字変数に割り当てる.**

● **書　式**　**LINE INPUT#〈ファイル番号〉,〈文字型変数〉**

● **文　例**　**LINE INPUT#1, A$**

● **解　説**　あらかじめOPEN文でINPUTモードを指定して開いておいたファイルからキャリッジリターンコード, ラインフィードコードで区切られている1行単位(最大 254文字)のデータを読み込み, 〈文字型変数〉で指定した変数に割り当てます.
〈ファイル番号〉はOPEN文で指定した番号です.
SAVE文でアスキー型式のプログラムファイルを作成した場合, 一行一行はキャリッジリターン(CHR$(13))とラインフィードコード(CHR$(10))とで区切られて書き出されています. またPRINT#文で作成したファイルにはPRINT#文を実行するごとに数値や文字列がキャリッジリターン+ラインフィードコードで区切られて書き出されています. LINE INPUT#文はこれらのファイルからキャリッジリターン+ラインフィードコードにはさまれているデータを読み込むときなどに使います.

サンプルプログラム

```
100 'テ゛ータ ノ ヨミコミ
110 OPEN "PASS" FOR INPUT AS #1
120 LINE INPUT #1,A$
130 PRINT A$
140 CLOSE

run
NATIONAL
Ok
```

★このプログラムを実行する前に, 41ページのプログラムを入力し実行しておいてください.

LOAD

ロード(コマンド)

● **機　能**　プログラムファイルをディスクからメモリへロードする.

● **書　式**　LOAD "<ファイルスペック>"〔,R〕

● **文　例**　LOAD "A：DEMO"

● **解　説**　<ファイルスペック>で指定されたファイルをメモリにロードします.
LOAD文を実行すると，LOAD文実行前にメモリ上にあったプログラムは消去されます．また，すべてのファイルは閉じられ変数は初期化されます.
<ファイルスペック>のデバイス名を省略すると，現在使用しているドライブからロードします.
Rオプションを指定すると，ファイルは開いたままでプログラムロード後，ただちに実行を開始します.

● **参　照**　SAVE

LOC

エルオーシー(関数)

● **機　能**　**ファイルの中の現在のレコード番号を調べる.**

● **書　式**　**LOC(〈ファイル番号〉)**

● **文　例**　**IF LOC(1)＞50　THEN END**

● **解　説**　ランダム入出力ファイルを指定したときLOCはGET文やPUT文で読み書きした最後のレコード番号の値になっています．ファイルを開いた直後はLOCは0になっています.

シーケンシャルファイルを指定したとき，LOCはそのファイルが開かれてから読み書きされたレコード数になっています.

ファイルがシーケンシャル入力モードで開かれた場合は，ベーシックは最初のセクタを読むので，そのファイルを読む前でもLOCの値は1になっています.

LOF

エルオーエフ(関数)

● **機　能**　**指定されたファイルの大きさを調べます.**

● **書　式**　**LOF(〈ファイル番号〉)**

● **文　例**　**IF M%＞LOF(1)　THEN PRINT "N.G"**

● **解　説**　〈ファイル〉で指定したファイルの実際の大きさをバイト単位で返します.

```
100 OPEN "PASS" AS #1
110 PRINT "File scale is"LOF(1)"bytes"
120 CLOSE

run
File scale is 11 bytes
Ok
```

★上のサンプルプログラムでは，41ページのデータファイルを調べていますが，他のページのプログラムも実行後調べてみてください.

(100行のファイル名を変更してください.)

LSET, RSET

エルセット，アールセット（ステートメント）

● **機　能**　ランダム入出力用バッファにデータを転送する．

● **書　式**　LSET <文字変数> = <文字式>
RSET <文字変数> = <文字式>

● **文　例**　LSET D$ = MKI$(D)

● **解　説**　<文字式>で示されるデータをFIELD文で指定した文字変数にセットします．<文字式>の文字がFIELD文で指定した文字より短かい場合は，LSET文では左詰め，RSET文では右詰めにされ，余分な所は空白で満たされます．逆にFIELD文で指定した文字より長い場合は，LSET文，RSET文ともに文字列の右側が切りすてられます．また数値データは取り扱えないので文字型に変換しなければなりません．
MKI$，MKS$，MKD$関数参照．
LSET文，RSET文はFIELD文で定義されていない文字型変数についても，そのフィールド内で右詰め，左詰めをするので，プリント出力の構成に利用できます．

● **参　照**　PUT，FIELD

MERGE

マージ（コマンド）

● **機　能**　メモリ上のプログラムとプログラムファイルとをマージ（混合）する．

● **書　式**　MERGE "<ファイルスペック>"

● **文　例**　MERGE "A：TEST"

● **解　説**　メモリ上のプログラムに<ファイルスペック>で指定したプログラムをマージ（混合）して，1つのプログラムにし，メモリ上に置きます．
<ファイルスペック>で指定するファイルはSAVE文を使ってアスキー形式でセーブされていなければなりません．

〈ファイルスペック〉でデバイス名を省略すると，現在使用しているドライブのアスキーファイルとマージします．

ファイルがカセットテープ上にある場合にのみ，ファイル名を省略することができます．

メモリ上のプログラムとファイルのプログラムに同一の行番号があった場合は，ファイルのプログラムが有効になります．

● 参 照　SAVE，LOAD

サンプルプログラム

```
10 'MERGE ノ サンプル
20 SCREEN 2
30 LINE(80,90)-(170,120),,BF
40 LINE(110,90)-(160,70),,BF
```

上記のプログラムをあらかじめアスキー形式でセーブしておきます．

〔例〕SAVE"SAMPLE",A

NEWと入力し，リターンキーを押した後，新たに下記のプログラムを入力します．

```
40 SCREEN 2
50 CIRCLE(150,120),13,1:PAINT(150,120),1
60 CIRCLE(100,120),13,1:PAINT(100,120),1
70 GOTO 70
```

その後，

〔例〕MERGE"SAMPLE"

を実行後，LISTを表示させると下記のようになります．

```
10 'MERGE ノ サンプル
20 SCREEN 2
30 LINE(80,90)-(170,120),,BF
40 LINE(110,90)-(160,70),,BF
50 CIRCLE(150,120),13,1:PAINT(150,120),1
60 CIRCLE(100,120),13,1:PPINT(100,120),1
70 GOTO 70
```

★上記のようにMERGEを実行すると，メモリ上の40行がファイルの40行に置き変わります．

MKI$, MKS$, MKD$

エムケイアイダラー, エムケイエスダラー, エムケイディーダラー (関数)

● **機　能**　数値を内部表現に対応した文字列に変換する.

● **書　式**　**MKI$ (<整数>)**
MKS$ (<単精度実数>)
MKD$ (<倍精度実数>)

● **文　例**　**A$＝MKS$ (26.4)**

● **解　説**　ランダムファイルは文字型データのみを扱います.
したがって数値型データは文字型データに変換して書き込み，読み込み時は，逆に文字型データを数値に変換します.
この場合の数値型データを文字型データに変換するのがMKI＄，MKS＄，MKD＄です.
MKI＄は整数を2バイトの文字列に変換します.
MKS＄は単精度実数を4バイトの文字列に変換します.
MKD＄は倍精度実数を8バイトの文字列に変換します.
これらの関数は数値をランダム入出力用バッファに対してLSET文, RSET文で書き込むときに使用します.
数値から文字への変換は数値が持つ内部表現の値をそのままそれに対応するキャラクタコードを割り当てます. この逆の動作をするのが, CVI, CVS, CVD関数です.
ご注意：STR＄関数とは，表現のしかたが異なります.

● **参　照**　CVI, CVS, CVD, LSET, RSET

**サンプル
プログラム**

```
100 'テ"ータ ノ カキコミ
110 OPEN "DATA5" AS #1
120 FIELD #1,2 AS D1$,4 AS D2$,8 AS D3$
130 INPUT "A(セイスウ)=";A%
140 INPUT "B(タンセイト")=";B!
150 INPUT "C(ハ"イセイト")=";C#
160 LSET D1$=MKI$(A%)
170 LSET D2$=MKS$(B!)
180 LSET D3$=MKD$(C#)
190 PUT #1,1
200 'テ"ータ ノ ヨミコミ
210 GET #1,1
220 A%=CVI(D1$)
230 B!=CVS(D2$)
240 C#=CVD(D3$)
250 PRINT "A=";A%
260 PRINT "B=";B!
270 PRINT "C=";C#
280 CLOSE
290 GOTO 100
```

```
run
A(セイスウ)=? 10
B(タンセイト")=? 12345.6789
C(ハ"イセイト")=? 123456789.123456
789
A= 10
B= 12345.7
C= 123456789.12346
A(セイスウ)=? ▒
```

NAME

ネーム(コマンド)

● **機　能**　ディスクファイルの名前を変更する.

● **書　式**　**NAME <旧ファイルスペック> AS <新ファイル名>**

● **文　例**　**NAME "A：ABC" AS "XYZ"**

● **解　説**　ファイル名の変更をします.
<旧ファイルスペック>は既に存在していなければなりません.
<新ファイル名>は既に存在していてはなりません.
ドライブ名を省略すると,現在使用しているドライブが選択されます.
NAME文はファイル名のみを変更し,その他は変化しません.
ご注意：ファイル名の変更をおこなうときは,CLOSEされた状態でなければなりません.

OPEN

オープン(ステートメント)

● **機　能**　ファイルを開く.

● **書　式**　**OPEN "<ファイルスペック>"〔FOR<モード>〕AS〔#〕<ファイル番号>〔LEN=<レコード長>〕**

● **文　例**　**OPEN "A：TEST" FOR INPUT AS#1**

● **解　説**　ファイルスペックで指定したファイルを開きます.
ファイルを使用するには必ずそのファイルを開いておかなければなりません.つまりPRINT#,INPUT$,PRINT# USING,INPUT#,LINE INPUT#,GET,PUT文の前に実行しなければなりません.

〈ファイルスペック〉でデバイス名を省略すると現在使用しているドライブが選択されます．

モード指定

FOR OUTPUT …新しくシーケンシャルファイルを作り，出力（ファイルへの書き込み）を行うことができる．

FOR INPUT ……既存のシーケンシャルファイルから入力（ファイルからの読み出し）を行うことを指示する．

FOR APPEND …既存のシーケンシャルファイルの後にデータを追加することを指示する．

省略………………ランダムファイルに対して入出力を行うことを指示します．

フロッピーディスクとして使用できる〈ファイル番号〉は1～6です．

2つ以上のファイルを同時にオープンする場合には，MAXFILES文で使用するファイル数を事前に定義する必要があります．また既にオープンされているファイルの番号は使えません．モードを省略，またはOUTPUTを指定してファイルにデータを出力した場合，同一名のファイルがあると，そのファイルは破壊され，同一名で新しいファイルを作成します．

〈レコード長〉はランダム入出力モードでファイルをオープンしたときの最大レコード長を1～256の範囲で指定します．省略すると256に設定されます．

既に開いているファイルは開くことができません．

ご注意：MAXFILESを省略したときオープンできるファイル番号は，1だけです．

それ以外は，Bad file numberエラーになります．

● 参　照　CLOSE

PRINT＃, PRINT＃ USING*

プリントシャープ,プリントシャープユージング(ステートメント)

● **機　能**　シーケンシャルファイルにデータを書き込みます.

● **書　式**　**PRINT＃〈ファイル番号〉,〔USING〈書式〉;〕〔〈式〉〔;〈式〉〕…〕**

● **文　例**　**PRINT＃1,USING "@＃＃＃＃.＃＃";A$;B**

● **解　説**　OPEN文でOUTPUTモードまたはAPPENDモードで開かれたシーケンシャルファイルにデータを書き込みます.
〈ファイル番号〉はOPEN文でOUTPUTモードまたはAPPENDモードで開かれたシーケンシャルファイルの番号です.
〈式〉はファイルに出力する数値式,または文字式です.
PRINT＃文は画面へ出力するものと同じものをファイルに書き込みます.(空白を詰めたり,切りすてたりしない.)そのためデータを後から読むときに正しく入力できるように書き込むデータを適切に区切る必要があります.
PRINT＃文およびPRINT＃ USINGの詳しい使用方法につきましては,「BASIC説明書」のPRINT＃およびPRINT USINGの説明を参照してください.

● **参　照**　INPUT＃

サンプル プログラム

```
100 CLS:PRINT "ｵﾜﾘ ﾊ END ﾄ ﾆｭｳﾘｮｸ"
110 OPEN "DATA6" FOR OUTPUT AS #1
120 'ﾃﾞｰﾀ ﾉ ｶｷｺﾐ
130 INPUT "ｼﾅﾓﾉ";N$
140 IF N$="END" THEN GOTO 190
150 INPUT "ﾈﾀﾞﾝ";P
160 INPUT "ｺｽｳ ";L%
170 PRINT #1,N$;",";P;L%
180 GOTO 120
190 'ｵﾜﾘ-1
200 PRINT:CLOSE
210 'ﾃﾞｰﾀ ﾉ ﾖﾐｺﾐ
220 T=0
230 OPEN "DATA6" FOR INPUT AS #1
240 IF EOF(1) THEN GOTO 300
250 INPUT #1,N$,P,L%
260 SUM=P*L%
270 PRINT N$;TAB(5);P;"*";L%;"=";SUM
280 T=T+SUM
290 GOTO 240
300 'ｵﾜﾘ-2
310 PRINT:PRINT "ｺﾞｳｹｲ...";T
```

```
run
ｵﾜﾘ ﾊ END ﾄ ﾆｭｳﾘｮｸ
ｼﾅﾓﾉ? ﾉｰﾄ
ﾈﾀﾞﾝ? 130
ｺｽｳ ? 3
ｼﾅﾓﾉ? ｴﾝﾋﾟﾂ
ﾈﾀﾞﾝ? 90
ｺｽｳ ? 6
ｼﾅﾓﾉ? END

ﾉｰﾄ    130 * 3 = 390
ｴﾝﾋﾟﾂ 90 * 6 = 540

ｺﾞｳｹｲ... 930
Ok
```

PUT

プット(ステートメント)

● 機　能　ランダム入出力用バッファ中のデータをランダムファイルへ出力する.

● 書　式　PUT〔#〕<ファイル番号>〔,<レコード番号>〕

● 文　例　PUT#1

● 解　説　<ファイル番号>で指定されたランダム入出力ファイルにランダム入出力用バッファの内容を書き込みます.
<レコード番号>を省略すると最後に実行したGET文, PUT文で指定した次のレコード番号が割り当てられます.
書き出すデータは, あらかじめFIELD文, LSET文, RSET文により準備しておかなければなりません.

● 参　照　GET

RUN

ラン(コマンド)

● 機　能　ディスク上のプログラムをロードし実行する.

● 書　式　RUN "<ファイルスペック>"〔,R〕

● 文　例　RUN "A：DEMO", R

● 解　説　ファイルスペックで指定されるプログラムファイルをロードし, ロード後ただちに実行します.
ディスク名を省略すると, 現在使用しているドライブからファイルをロードします.
ファイル名を省略することはできません.
RUN文はすべてのファイルを閉じ, 変数を初期化し, RUN文の実行前にメモリ上にあったプログラムをこわします.
Rオプションを指定するとすでに開かれているファイルがあれば, ファイルは開かれたままプログラムを実行します.
DISK BASICにおけるRUN文は通常のベーシックのRUN機能を完全に包含しています.

SAVE

セーブ(コマンド)

● **機　能**　**メモリ上のプログラムをファイルにセーブします.**

● **書　式**　**SAVE"<ファイルスペック>"〔, A〕**

● **文　例**　**SAVE "A：DEMO", A**

● **解　説**　メモリ上のプログラムを<ファイルスペック>で指定されたプログラムファイルへセーブします.

指定した<ファイルスペック>のファイルが無い時は新しいファイルを作成し，既にある場合は内容を書きかえます.

Aオプションが指定された場合には，プログラムはアスキー形式でセーブされます．省略された場合，中間言語にデータを圧縮してセーブします.

MERGE文やその他プログラムファイルを操作するにはアスキー形式でセーブする必要があります.

ご注意：カセットテープの場合，中間言語でセーブするとき(CSAVE）とアスキー形式でセーブするとき(SAVE)とでは命令が異なりましたが，ディスクにセーブする場合はどちらもSAVE文でAオプションで使い分けるようになっていますので注意する必要があります.

● **参　照**　**MERGE, LOAD**

SYSTEM

システム(コマンド)

● **機　能**　**DISK BASICから制御をMSX-DOSに戻す.**

● **書　式**　**{ CALL SYSTEM | _SYSTEM }**

● **文　例**　**CALL SYSTEM**

● **解　説**　この命令はMSX-DOSシステムディスク(MSXDOS. SYSとCOMMAND. COM)をディスクドライブに装着し, MSX-DOSからDISK BASICを起動した場合のみ有効です.
MSX-DOSに戻る前にすべてのファイルは閉じられ, メモリは初期化されます.

第3章

サンプルプログラム

DISK BASICのコマンド，ステートメント，関数を使って，簡単な「予定表」のプログラムを作成してみましょう．

データとしては，月，日，時間，内容を入力し，表示は全データまたは，指定日のみの表示が可能なものにします．

1．よていひょう1…シーケンシャルファイル方式

●60ページにプログラムを，61ページにフローチャートを掲載しています．
　また，フローチャートの各ブロックごとに，(　)で該当するプログラム行を記しています．

「よていひょう1」のプログラムは，ファイルへのデータ転送を220行で実行するようになっていますが，その内容は次のようになります．

(ØD, ØAはCR, LFコードで220行の命令実行時に自動的に挿入されます．)

〔例〕DT＄＝"Ø5/1Ø"，TM＄＝"13：ØØ"，NIYU＄＝"うちあわせ"

Ø	5	/	1	Ø	,	1	3	:	Ø	Ø	,	う	ち	あ	わ	せ	ØD	ØA

1レコード

書き込まれるデータフォーマットの詳細は，お手持ちのBASIC説明書のPRINT＃の項を参照してください．

〈操作方法〉

●データの入力は，画面との対話形式でおこないます．
　(日付は必ず☒☒/☒☒の形式で入力してください．)

●データの終了は，日付入力時点でEキーを押した後，リターンキーを押してください．

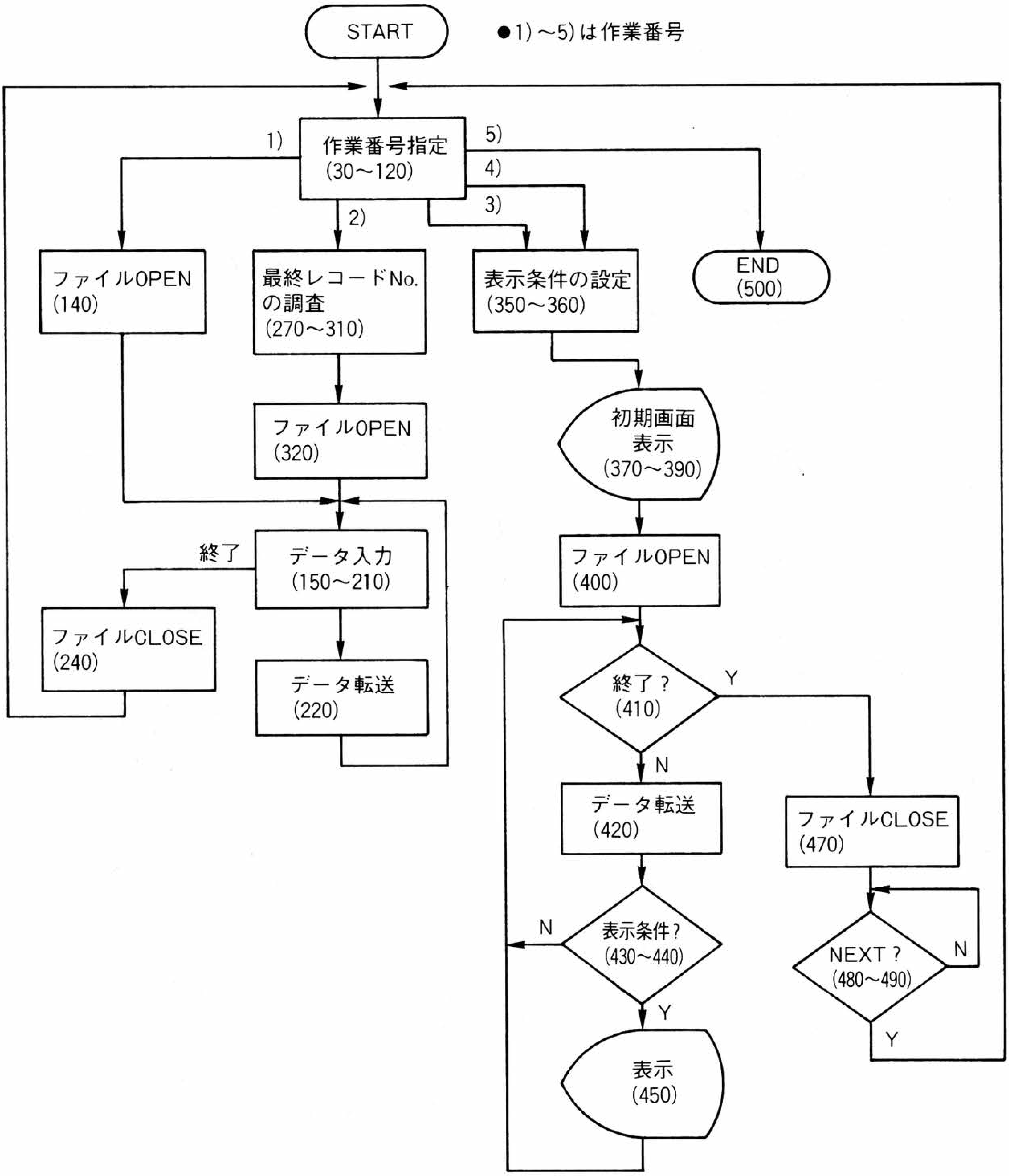
●表示の指定は全データまたは指定日のみの2通りを指定することができます．
　(データの並べ変えは，おこなっていません．)

●書き込まれたデータの修正や削除の機能はありません．
　(新規作成をおこなうと，旧データは，すべて消去されます．)

●よていひょう1プログラム

```
10 'よていひょう---1
20 'さぎょうばんごう の しじ
30 CLS
40 LOCATE 6,2:PRINT"*** よていひょう ***"
50 LOCATE 3,4:PRINT"1) しんき さくせい"
60 LOCATE 3,6:PRINT"2) データ ついか"
70 LOCATE 3,8:PRINT"3) データ ひょうじ---ぜん データ"
80 LOCATE 3,10:PRINT"4) データ ひょうじ---してい日のみ"
90 LOCATE 3,12:PRINT"5) おわり"
100 LOCATE 1,15:INPUT"さぎょうばんごう にゅうりょく";N
110 IF N<1 OR 5<N THEN 30
120 ON N GOTO 130,260,350,360,500
130 'しんき さくせい
140 OPEN "YOTEI.DAT" FOR OUTPUT AS #1
150 CLS
160 LOCATE 0,3:INPUT"月/日(XX/XX)OR E";DT$
170 IF DT$="E" OR DT$="e" THEN 240
180 LOCATE 0,5:INPUT "時/分(XX:XX)";TM$
190 LOCATE 0,7:INPUT "ないよう(MAX 15じ)";NIYU$
200 LOCATE 0,9:INPUT "OK?(Y/N)";ED$
210 IF ED$="Y" OR ED$="y" THEN 220 ELSE 150
220 PRINT #1,DT$;",";TM$;",";NIYU$
230 GOTO 150
240 CLOSE #1
250 GOTO 30
260 'データ ついか
270 OPEN "YOTEI.DAT" FOR INPUT AS #1
280 IF EOF(1)=-1 THEN 310
290 INPUT #1,DT$,TM$,NIYU$
300 GOTO 280
310 CLOSE #1
320 OPEN "YOTEI.DAT" FOR APPEND AS #1
330 GOTO 150
340 'データ ひょうじ
350 D1$="すべて":GOTO 370
360 LOCATE 1,17:INPUT "ひょうじ月/日(XX/XX)";D1$
370 CLS
380 PRINT D1$;" のよていひょう"
390 PRINT" 月/日    時/分      ないよう"
400 OPEN "YOTEI.DAT" FOR INPUT AS #1
410 IF EOF(1)=-1 THEN 470
420 INPUT #1,DT$,TM$,NIYU$
430 IF N=3 THEN 450
440 IF DT$<>D1$ THEN 410
450 PRINT DT$;" ";TM$;" ";NIYU$
460 GOTO 410
470 CLOSE #1
480 PRINT"データ しゅうりょう"
490 INPUT "HIT ANY KEY";Y$:GOTO 30
500 CLS:END
```

●よていひょう1フローチャート



2．よていひょう2…ランダムファイル方式

●63ページにプログラムを，65ページにフローチャートを掲載しています．
また，フローチャートの各ブロックごとに，（　）で該当するプログラム行を記しています．

「よていひょう1」では，データの一部削除はできませんでしたが，指定した月，日のデータ削除ができる機能を追加しています．

このために，ファイルとしてランダムファイル＃1，＃2の2つのファイルを使用します．

＃1は，各レコードの有効性を管理するテーブル，＃2は，データ書き込み用です．

＃1は，250バイトのデータのそれぞれが，各レコードのフラグとなります．

該当レコードが「空」のときは"Ø"，データが入っているときは"1"となります．

ファイル内のデータは，次のように書き込まれます．(＃2)

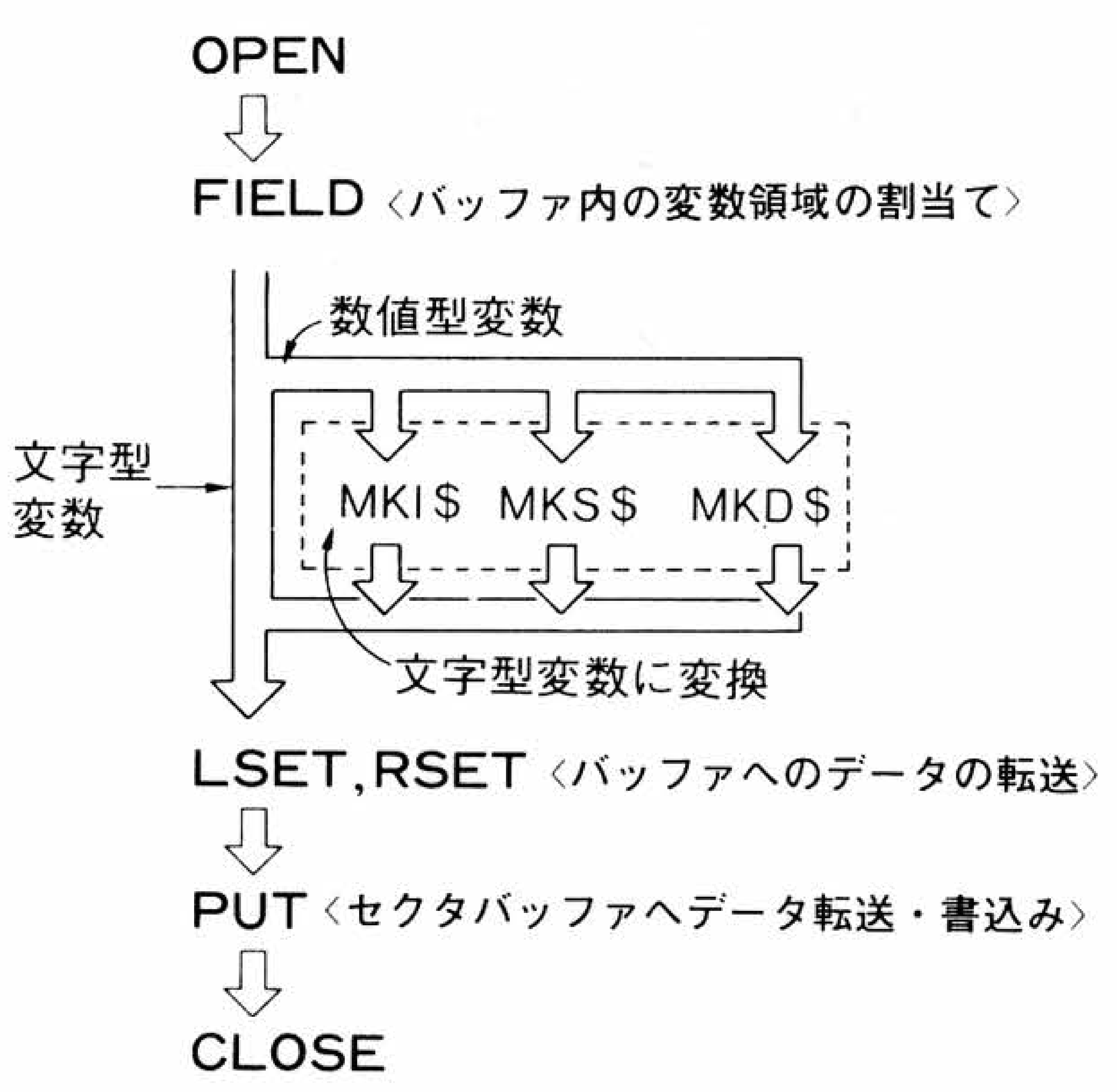
レコードN	Ø	7	/	2	1	1	3	:	Ø	Ø	は	ん	は	゛	い	か	い	き	゛			
レコードN+1	Ø	7	/	2	1	1	5	:	Ø	Ø	M	し		ら	い	し	ゃ					
レコードN+2	Ø	7	/	2	1	1	8	:	Ø	Ø	テ	ニ	ス									

月／日　　時：分　　内容(15字以内)

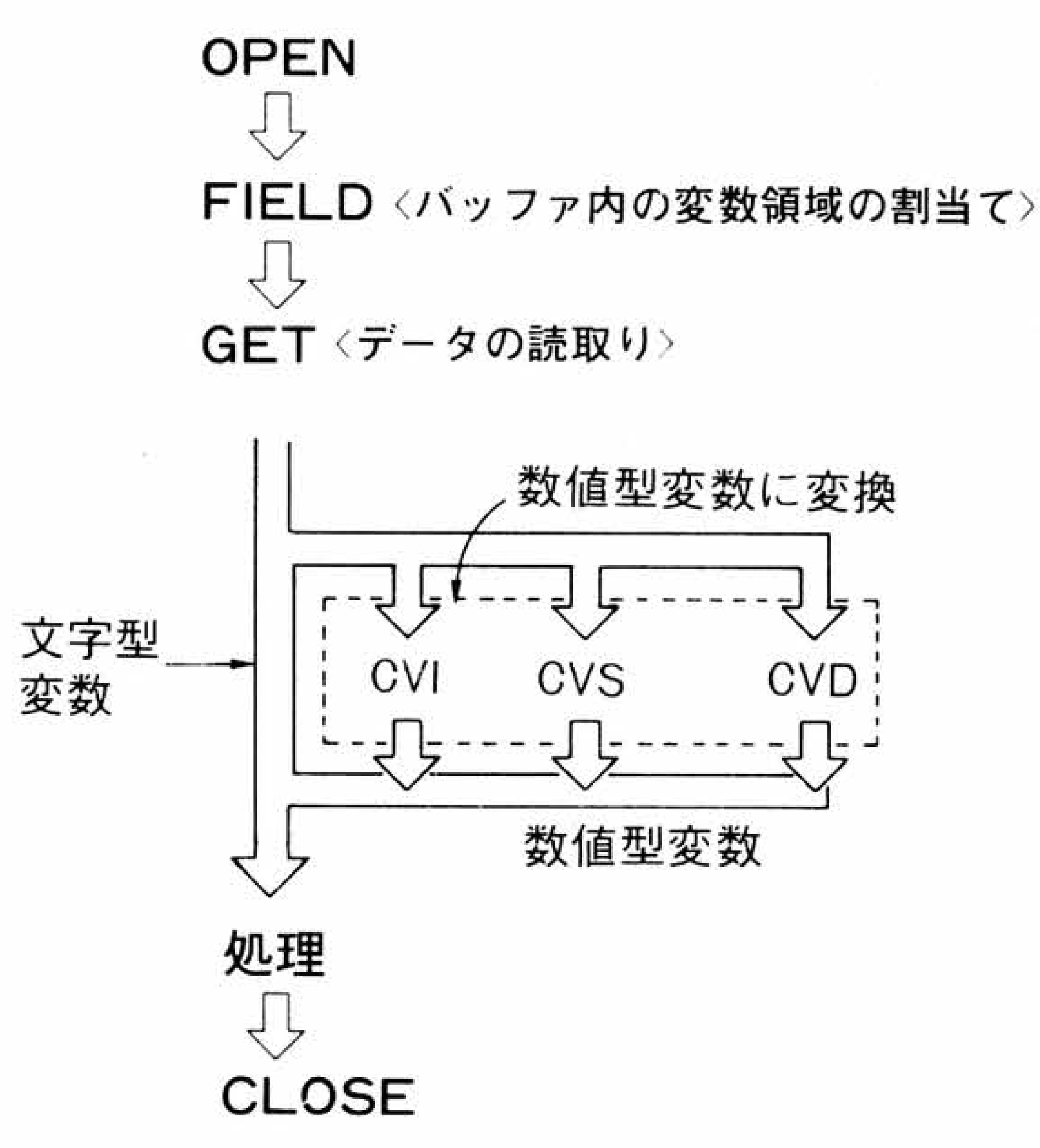
操作方法は，データの一部削除機能が追加された以外は，「よていひょう1」と同様におこなってください．

ランダムファイルへの書き込み，読み込みについて，全体の流れを下に記しますので参考にしてください．

●書き込み



●読み込み



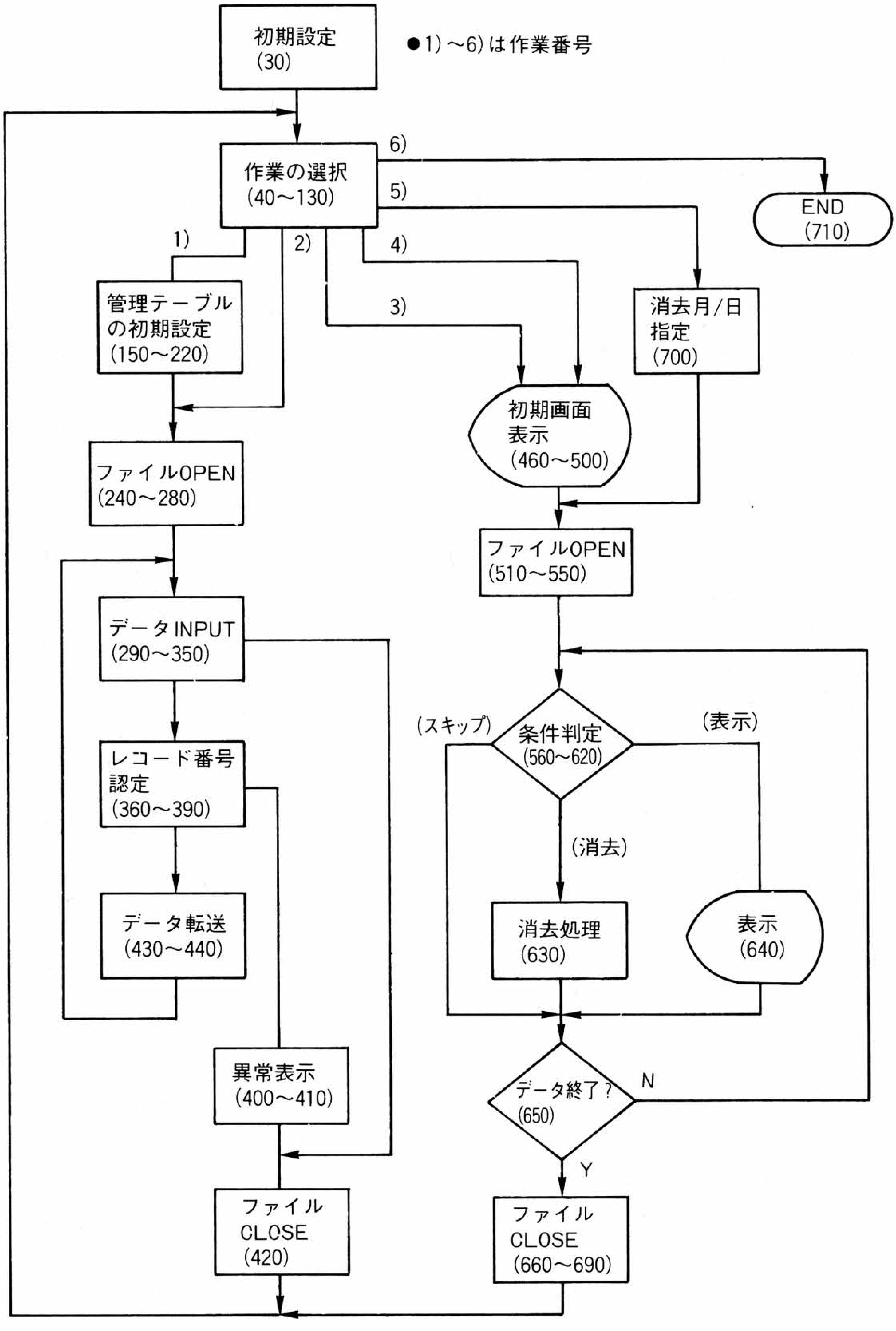
●よていひょう2プログラム

```
10 'よていひょう---2
20 'さぎょうばんごう の しじ
30 CLEAR 800:CLS:MAXFILES=2
40 LOCATE 6,2:PRINT"*** よていひょう ***"
50 LOCATE 3,4:PRINT"1) しんき さくせい"
60 LOCATE 3,6:PRINT"2) データ ついか"
70 LOCATE 3,8:PRINT"3) データ ひょうじ---ぜん データ"
80 LOCATE 3,10:PRINT"4) データ ひょうじ---してい日のみ"
90 LOCATE 3,12:PRINT"5) データ いちぶじょうきょ"
100 LOCATE 3,14:PRINT"6) おわり"
110 LOCATE 1,17:INPUT"さぎょうばんごう にゅうりょく";N
120 IF N<1 OR 6<N THEN 30
130 ON N GOTO 140,240,460,470,700,710
140 'しんき さくせい
150 OPEN "FAT1.DAT" AS #1
160 FIELD #1,250 AS FAT1$
170 FF$=""
180 FOR F=1 TO 250
190 FF$=FF$+"0"
200 NEXT F
210 LSET FAT1$=FF$:PUT #1,1
220 CLOSE #1
230 'データ ついか
240 OPEN "FAT1.DAT" AS #1
250 FIELD #1,250 AS FAT1$
260 GET #1,1
270 OPEN "YTI.DAT" AS #2 LEN=25
280 FIELD #2,5 AS FDT$,5 AS FTM$,15 AS NN$
290 CLS
300 LOCATE 0,3:INPUT"月/日(XX/XX) OR E";DT$
310 IF DT$="E"OR DT$="e"  THEN 420
320 LOCATE 0,5:INPUT "時/分(XX:XX)";TM$
330 LOCATE 0,7:INPUT "ないよう(MAX 15じ)";NIYU$
340 LOCATE 0,9:INPUT "OK?(Y/N)";ED$
350 IF ED$="Y" OR ED$="y" THEN 360 ELSE 290
360 FOR C1=1 TO 250
370 F1$=MID$(FAT1$,C1,1)
380 IF F1$="0" THEN MID$(FAT1$,C1,1)="1":GOTO 430
390 NEXT C1
400 PRINT"  DATA AREA OVER FLOW!!"
410 FOR C2=0 TO 1500:NEXT C2
420 LSET FAT1$=FAT1$:PUT#1,1:CLOSE #1:CLOSE #2:GOTO 10
430 LSET FDT$=DT$:LSET FTM$=TM$:LSET NN$=NIYU$
440 PUT #2,C1:GOTO 290
```

↓・64ページに続く

```
450 'すへ"ての テ"ータ ひょうし"
460 D1$="すへ"て":GOTO 480
470 LOCATE 1,19:INPUT "ひょうし"月/日(XX/XX)";D1$
480 CLS
490 PRINT D1$;" のよていひょう"
500 PRINT" 月/日    時/分      ないよう"
510 OPEN "FAT1.DAT" AS #1
520 FIELD #1,250 AS FAT1$
530 GET #1,1
540 OPEN"YTI.DAT" AS #2 LEN=25
550 FIELD #2,5 AS DT$,5 AS TM$,15 AS NIYU$
560 FOR C1= 1 TO 250
570 F1$=MID$(FAT1$,C1,1)
580 IF F1$="0" THEN 650
590 GET #2,C1
600 IF N=3 THEN 640
610 IF DT$<>D1$ THEN 650
620 IF N=4 THEN 640
630 MID$(FAT1$,C1,1)="0":GOTO 650
640 PRINTDT$;" ";TM$;" ";NIYU$
650 NEXT C1
660 LSET FAT1$=FAT1$:PUT #1,1
670 CLOSE #1:CLOSE #2
680 PRINT"DATA END"
690 INPUT "HIT ANY KEY";Y$:GOTO 30
700 LOCATE 1,19:INPUT"しょうきょ月/日(xx/xx)";D1$:GOTO 510
710 CLS:END
```

●よていひょう２フローチャート



第 4 章

MSX-DOSの概要

4.1 MSX-DOSとは

MSXシステム(パーソナルコンピュータ)用のDOS(Disk Operating System)のことです.

これはユーザーがディスクを幅広く，また効率よく使用できるようにするためのシステムプログラムです.

MSX-DOSは,MS-DOSとほぼ同様の機能を持つMSXシステム用のDOS(Disk Operating System)です.

(MS-DOSあるいはDOSについて書かれた書籍などを参考にされると,より理解が深まると思われます.)

4.1.1 MSXDOS. SYS/COMMAND. COM

MSX-DOSシステムは，MSXDOS. SYSとCOMMAND. COMと呼ばれる2つのプログラムを必要とします.

MSXDOS. SYSとは,MSX-DOSの核となるもので,COMMAND. COMとは，コマンドを実行するプログラムです.

4.1.2 システムフロッピーディスクの作成

MSXDOS. SYSおよびCOMMAND. COMは,付属のフロッピーディスクに組み込まれています.

MSX-DOSを使用される場合には，次のような操作でシステムフロッピーディスクを作成して，ご使用ください.

1. フォーマット済みのフロッピーディスクを準備します.
   (「1.3フォーマットについて」を参照してください.)
2. DISK BASICを起動します.
3. 付属のフロッピーディスクの下記2つのファイルを，DISK BASICのCOPYコマンドで，準備したフロッピーディスクにコピーします.
   (「1.6ディスクコピーのしかた」を参照してください.)

   MSXDOS. SYS　　COMMAND. COM

注　誤操作により，付属のフロッピーディスクの内容をこわすことのないよう，付属のフロッピーディスクは，ライトプロテクト状態にしてコピーをおこなってください.(9ページ参照)

上記2つのファイルのコピーが終了すれば，MSX-DOSシステムフロッピーディスクの作成は完了です.

4.2 MSX-DOSの使いかた

システムフロッピーディスクによりMSX-DOSを起動すると，下のような初期画面があらわれます．しかし，この状態ではまだMSX-DOSはスタートしていません．

```
MSX-DOS version X.XX
Copyright 1984 by Microsoft

COMMAND version X.XX

Current date is Sun 84-01-01
Enter new date:▒
```

●ご使用のフロッピーディスクにMSX-DOSの自動実行バッチファイル（77ページ）がある場合には，「4.1.2システムフロッピーディスクの作成」を参照し，システムディスクを作成して操作してください．

4.2.1 年―月―日

上記表示に対して，日付変更の必要がない場合は，リターンキーを押します．

```
MSX-DOS version X.XX
Copyright 1984 by Microsoft

COMMAND version X.XX

Current date is Sun 84-01-01
Enter new date:
A>▒
```

日付を変更するときは，

〈**年**〉―〈**月**〉―〈**日**〉

と入力し，リターンキーを押します．

●年,月,日はそれぞれハイフォン(―)またはスラッシュ(/)でつなぎます．
(詳しくは79ページ参照)

●MSX2 BASICの機種では，内蔵の時計から日付と時刻を読み込みますので，この操作はありません．

4.2.2 A>とは

日付の入力を終えると画面上に

A>■

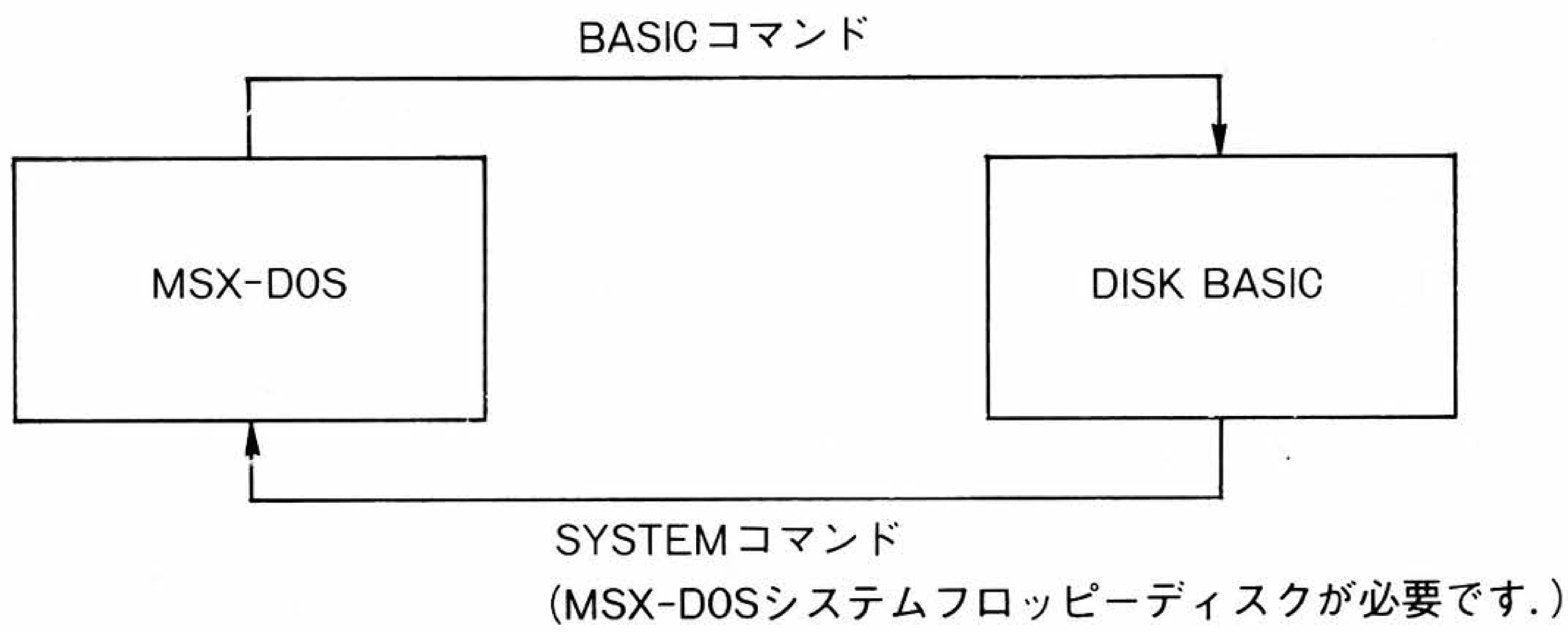
と表示されます．

これをAプロンプトと呼びMSX-DOSが，入力待ち状態であることを示しています．

4.2.3 BASICコマンドとCALL SYSTEMコマンド

MSX-DOSのコマンドにBASICがあります.
このコマンドを使用するとMSX-DOSの状態からDISK BASICの状態へ移ることができます.(76ページ参照)

また, MSX-DOSの状態からDISK BASICの状態へ移ったときは, DISK BASICのSYSTEMコマンドにより再びMSX-DOSの状態にもどることができます.(MSX-DOSシステムフロッピーディスクが必要です.)



4.3 DIR

DIRコマンド(80ページ参照)は、DISK BASICのFILESコマンドと同じ働きをし、ディスク上のファイルを画面に表示します.

ディスクを装着した状態で

A>DIR (表示されているA>に続けてDIRを入力)

と入力し、リターンキーを押すと、ディスク上のファイルが下のように表示されます.

```
ファイル名   拡張子   ファイルの大きさ   作成した日付*
MSXDOS      SYS      2432        84-01-01
COMMAND     COM      6528        84-01-01
             2 files
      719872 bytes free
  A>■
```

下段に表示されている内容は, 画面上に表示されたファイル数および現在の残りバイト数です.(上記の例は, システムフロッピーディスクにユーザープログラムをセーブする前の状態のものです.)

*ただしMSX 1の機種で, 電源を入れた後日付を入力せずに作成したファイルの日付は, 84−01−01となります.

4.4 ディスクコピーのしかた

フロッピーディスクは，取扱いが便利なうえディスク操作も簡単なために，ちょっとした不注意でディスク上のプログラムやデータをこわしてしまうことがあります．

大切なプログラムやデータは，もう一枚のフロッピーディスクにコピーしておかれることをお奨めします．(このことをバックアップと言います．)

ここでは，一台のディスクドライブでコピーをする方法を説明します．

1．コンピュータをMSX-DOSを使用する状態にします．

2．作成したプログラムやデータの入ったフロッピーディスクをディスクドライブに装着します．

3．A＞の後にCOPY　A：〈ファイル名〉.〈拡張子〉　B：と入力し，リターンキーを押します．

●フロッピーディスクからプログラムやデータをメモリーへロードし画面に表示した後，次のようなメッセージが返ってきます．

Insert diskette for drive B：
and strike a key when ready ■

4．ここで，ディスクドライブからプログラムやデータの入ったフロッピーディスクを抜き取り，コピー用の別のフロッピーディスクを装着します．

5．リターンキーを押します．

ディスクドライブインジケータが点灯し，「コピー」が始まります．

●コピーが完了したときは次のように表示されます．

1 File(s) copied
A＞■

■ディスク上の全てのファイルをコピーするとき

70ページ1.2.の操作をおこなった後、以下の操作をおこなってください.

3. A>の後にCOPY　A：＊.＊　B：
   と入力し、リターンキーを押します.
   ●フロッピーディスクから数ファイルメモリーへロードし画面に表示した後、次のようなメッセージが返ってきます.
   Insert diskette for drive B：
   and strike a key when ready ■
4. ここで、ディスクドライブからプログラムやデータの入ったフロッピーディスクを抜き取り、コピー用の別のフロッピーディスクを装着します.
5. リターンキーを押します.
   ディスクドライブインジケータが点灯し、「コピー」が始まります.
   ●数ファイル分のコピーが終ると
   Insert diskette for drive A：
   and strike a key when ready ■
   と表示されます.
6. コピー用のフロッピーディスクを抜き取り、プログラムやデータの入ったフロッピーディスクを装着します.
7. リターンキーを押します.
   ●再び、
   Insert diskette for drive B：
   and strike a key when ready ■
   と表示されます.

大きなファイルをコピーするときは、4.5.6.7.の操作を「××File(s)Copied」が表示されるまで、くり返しおこなってください.

4.5 カーソル移動キーの使いかた

MSX-DOSモードでは，カーソル移動キーは以下のような機能があります.

▲キー：現在入力中のA＞以降の文字列を全て消去します.
カーソルは，A＞の次の文字位置に移動します.

▼キー：前回実行したA＞以降の文字列を表示します.
現在すでに入力された文字列(N文字)がある場合には，その文字列に続けて，前回実行した文字列のN＋1文字目以降の文字列を続けて表示します.

例） 前回実行の文字列　A>COBY A：SAMPLE B：
現在の入力行　A>COP
↓ ▼ キーを押す
▼ キーを押した後の入力行　A>COPY A：SAMPLE B：

◀キー：カーソルの直前の文字を消去し，カーソルを消去した文字の位置に移動します.
ただしA>は消えません.

▶キー：前回実行したA＞以降の文字列を1文字ずつ表示します.
表示する文字の対応は，▼ キーの場合と同様です.

4.6 エラー対応

操作を間違えた場合など，次のような表示が現われることがあります.

Abort，Retry，Ignore？■

このような場合には，A，R，Iを入力するとそれぞれ次のような動作をします.

Aを入力すると…プログラムの実行を中止します.
(A＞が表示され入力待ちの状態になります.)

Rを入力すると…エラーが発生した命令を再度実行します.

Iを入力すると…エラーが発生した命令を無視して，次に進みますので誤った処理になります.
通常は使用しないでください.

第 5 章

MSX-DOSリファレンス

MSX-DOSのコマンド

第5章の見方

● **機　能**　命令の機能を簡単に示します.

● **書　式**　命令の書き方を示します. 実際の入力時には次のような決まりに従ってください.

- ●英字の大文字で示された項目は, そのとおり入力します.
  入力する場合は, 小文字でも大文字でもかまいません.
- ●カギカッコ〈　〉で囲まれた項目は, ユーザーが指定します.
- ●角カッコ〔　〕で囲まれた項目は, オプションであり, 省略することができます. 省略した場合, 省略値(MSX-DOSによって設定される値)または以前に指定した値が適用されます.
- ●コマンドと記述部(ファイルスペックなど)の間は, 必ず1字文以上のスペースを空けてください.
  (書式中の␣はスペースの意味です.)

BASIC

ベーシック

● **機　能**　**DOSからBASICを起動する.**

● **書　式**　**BASIC␣〔<ファイルスペック>〕**

● **解　説**　MSX-DOSが起動されている状態からDISK BASICを起動するためには, BASICコマンドを実行します.
<ファイルスペック>によってBASICのプログラムファイルを指定すると, BASICを起動した後, そのプログラムをロードし実行を開始します.

＊BASICコマンドの実行により内部のスロットが切換えられ, BASIC ROMが有効となります. したがって, MSX-DOSとDISK-BASICではメモリ構成が変化します.

COPY

コピー

● **機　能**　**<ファイルスペック1>を<ファイルスペック2>へコピーする.**

● **書　式**　**COPY␣<ファイルスペック1>␣〔<ファイルスペック2>〕**

● **解　説**　<ファイルスペック1>のファイルを<ファイルスペック2>はコピーします.
複数のファイルを同時にコピーすることもできます.
(詳しくは「**4.4** ディスクコピーのしかた」を参照してください.)

COPYコマンドを応用した自動実行バッチファイルの作成方法

■自動実行バッチファイル(AUTOEXEC.BATファイル)の作りかた

自動実行バッチファイル(AUTOEXEC.BATファイル)とは，MSX-DOSを起動するたびにBASICを自動的にロードし，あらかじめ作成しておいたプログラムを実行させる事ができるファイルです．

操作は次のようにおこないます．
(例としてあらかじめ作成しておいたプログラムのファイル名をPRG 1とします．)

1．COPY CON AUTOEXEC.BATと入力し，リターンキーを押してください．

●このメッセージを入力すると，MSX-DOSは本体から自動実行バッチファイル(AUTOEXEC.BATファイル)へ，データをコピーするように命令します．

```
A>COPY CON AUTOEXEC.BAT
■
```

2．BASIC PRG 1と入力し，リターンキーを押してください．

●このメッセージは，自動実行バッチファイル(AUTOEXEC.BATファイル)に入っていきます．
これでMSX-DOSが起動するたびに，BASICをロードしプログラムPRG 1を実行する命令がMSX-DOSに与えられます．

```
A>COPY CON AUTOEXEC.BAT
BASIC PRG 1
■
```

3．コントロールキーを押しながら[Z]キーを押してください．

```
A>COPY CON AUTOEXEC.BAT
BASIC PRG 1
^Z■
```

4．リターンキーを押してください．

●これでBASIC PRG 1と言うコマンドが，自動実行バッチファイル(AUTOEXEC.BATファイル)に入ります．

これでMSX-DOSを起動するたびに，プログラムPRG 1が自動的に実行されます.

あらかじめ作成したプログラムのファイル名の他に，MSX-DOSコマンドや連続したコマンドなども組み込むことができます.

ご注意： 自動実行バッチファイルは，電源を入れた時，日付入力が表示されません.
あらかじめ自動実行バッチファイルの中にDATE コマンドを入れておかれる事を，お奨めします.

DATE

デート

● **機　能**　日付を表示し，日付の変更を求める．

● **書　式**　DATE␣〔〈年－月－日〉〕

● **解　説**　パラメータなしで入力した場合，DATEコマンドはつぎのメッセージを表示します．

Current date is 曜日 yy －mm －dd

Enter new date：_

表示された日付を変更する必要がない場合には，RETURNキーを押します．
DATEコマンドのパラメータとして，以下のように日付を入力するとメッセージは表示されずに変更が行われます．

DATE　81-3-9

日付の入力には，数字だけが使用できます．(文字は許されていません．)
使えるパラメータの範囲は．以下のとおりです．

〈yy〉＝80～99または1980～2099

〈mm〉＝ 1 ～12

〈dd〉 ＝ 1 ～31

年，月，日の入力は，ハイフン(—)かスラッシュ(／)で区切ります．MSX-DOSは，月の大小やうるう年の場合でも，正しく判別するようにプログラミングされています．
パラメータや区切り記号が正しくない場合，MSX-DOSは次のメッセージを表示します．

Invalid date

Enter new date：_

このメッセージは，正しい入力がなされるまで繰り返されます．

DEL

デリート

● **機　能**　**〈ファイル指定〉で指定されたファイルをすべて消去する．**

● **書　式**　**DEL␣〈ファイル指定〉**

● **解　説**　ファイル指定が＊．＊(ワイルドカード)の場合は，プロンプト"Are you sure ？(Y/N)"が表示されます．"Y"を入力すると，要求どおりにすべてのファイルが消去されます．

(ワイルドカードについては，「1.4.5 ワイルドカードとは」16ページ参照)

●ERASEも，DELコマンドと同じです．

DIR

ディレクトリー

● **機　能**　**ディレクトリー内のファイルをリストアップする．**

● **書　式**　**DIR␣〔〈ファイルスペック〉〕〔〈/P〉〕〔〈/W〉〕**

● **解　説**　パラメータがない場合(DIR)，デフォルトドライブ上にあるディレクトリーの項目をすべて表示します．ドライブ指定のみがある場合(DIR〈d：〉)は，指定されたドライブのディスク上にある全てのファイルディレクトリーを表示します．拡張子なしのファイル名だけがある時(DIR〈ファイル名〉)，デフォルトドライブのディスク上の，指定されたファイル名を持つすべてのファイルを表示します．完全なファイルスペックが指定された場合(DIR〈ファイルスペック〉)，指定されたドライブにあるディスクの，指定されたファイルに関する情報を表示します．いずれの場合においても，ファイルは，バイト単位でのサイズ，最後に修正を受けた日付とともに表示されます．(ただしDATEを設定をしていないときは，デフォルトの日付が表示されます．)

ファイル名のパラメータとして，クエスチョンマーク(?)とアスタリスク(*) のワイルドカードが使用できます．

DIRコマンドの呼び出しで，同じ意味を持つものは，

コマンド	同等のコマンド形式
DIR	DIR␣*.*
DIR␣FILE	DIR␣FILE.*
DIR␣.EXT	DIR␣*.EXT
DIR␣.	DIR␣*.

になります．

DIRでは2つのスイッチが使用できます．

／Pスイッチは，ページモードを選択します．／Pの付いた場合は，スクリーンがいっぱいになったところで表示を止めます．再開するには，任意のキーを押します．

／Wスイッチは，ワイドディスプレイを選択します．／Wが付いた場合は，ファイル名のみを表示し，他の情報は表示しません．ファイル名は1行に表示できるだけ表示します．

ERASE

イレーズ

● **機　能**　**<ファイル指定>で指定されたファイルをすべて消去する.**

● **書　式**　**ERASE␣<ファイル指定>**

● **解　説**　DELコマンドと同じです.

FORMAT

フォーマット

● **機　能**　**MSX-DOSで使えるように，指定されたドライブのディスクをフォーマット(初期化)する.**

● **書　式**　**FORMAT␣〔<d：>〕**

● **解　説**　指定されたドライブのディスクを初期化します.
画面に表示されるガイドメッセージに従って，操作してください.
(操作手順は，DISK BASICのFORMATコマンドと同様です.)

MODE

モード

● 機　能　画面に表示する桁数を設定する.

● 書　式　MODE␣〈桁数〉

● 解　説　〈桁数〉の範囲は1〜40までです.
1〜32は画面モード1，33〜40は画面モード0になります.
(初期値は29に設定されています.)

PAUSE*

ポーズ

● 機　能　バッチファイルの実行を一時停止する.

● 書　式　PAUSE␣〔〈コメント〉〕

● 解　説　バッチファイルの実行中にディスクの交換をしたり，何かほかのことを行うことが必要になる場合があります．PAUSEコマンドはこのような目的のためのものです.PAUSEコマンドはCTRL+C以外のキーを押すまで実行を停止します.
PAUSEコマンドを実行すると,

Strike a key when ready ■

と表示される.このときCTRL+C以外のキーを押すと，処理は再開されます.CTRL+Cを押すと,以下のようなプロンプトが表示されます.

Terminate batch file (Y/N) ?

ユーザーがこのプロンプトに対して"Y"を入力すると，バッチ処理の残りは打切られ，制御はオペレーティングシステムのコマンドレベルに戻ります．従って，PAUSE コマンドはバッチファイルをいくつかに分割するのに使用でき，ユーザーは途中でその処理を終了させることもできます．オプションのコメント(メッセージ)を，PAUSEと同じ行に入力することにより,バッチファイルのユーザーに指示を与えることもできます.

REM*

リマーク

● **機　能**　**バッチファイルの実行がこのコマンドまでくると, REMとおなじ行に示されているコメントを表示する.**

● **書　式**　**REM␣〔<コメント>〕**

● **解　説**　REMは上記の動作以外に, 他に何の影響も与えません. REMコマンドと後に続くコメントとの間は, スペース, タブ, カンマのいずれかで区切らなければなりません.

REN

リネーム

● **機　能**　**第1パラメータの<ファイルスペック>(ファイル名)を, 第2パラメータ(新しいファイル名)に変更する.**

● **書　式**　**REN␣<ファイルスペック>␣<(新しい)ファイル名>**

● **解　説**　デフォルトドライブ以外に入っているファイルの名前を変更する場合は, 第1パラメータ(ファイルスペック)のドライブ指定が必要です. 第2パラメータ(新しいファイル名) のドライブ指定はすべて無視され, ファイルは同じディスク上に残ります.

パラメータの指定にはクェスチョンマーク(?), アスタリスク(*)のワイルドカードを使用することができます. この時, 第1パラメータ (ファイルスペック) で指定したファイルと第2パラメータで指定したファイルとの間で, ファイル名の各文字が1対1に対応して処理されます.

RENAMEは, RENコマンドと同じです.

(ワイルドカードについては,「1.4.5 ワイルドカードとは」16ページ参照)

TIME

タイム

● **機　能**　**時刻を表示し，時刻の変更を求める．**
タイマー(リアルタイムロック)機能を有している場合のみ使用できます．

● **書　式**　**TIME␣〔〈hh〉〔：〈mm〉〕〕**

● **解　説**　パラメータなしでTIMEコマンドを入力した場合，以下のメッセージが表示されます．

Current time is 〈hh〉：〈mm〉：〈ss.ss〉X

Enter new time：

表示された時刻を変更する必要がない場合は，RETURNキーだけを押します．
以下のように，オプションでTIMEコマンドのパラメータとして，データを与えることができます．

TIME 8：20

新しい時刻の入力には，数字だけが使用できます．(文字は許されていません．)
使えるパラメータの範囲は，以下のとおりです．

〈hh〉＝00～23

〈mm〉＝00～59

時，分は，コロン(：)で区切って入力します．
TIMEコマンドは，DATEコマンドと共通に管理されています．TIME(時刻)によって，DATE(日付)が自動的にコントロールされます．
MSX-DOSは，パラメータと区切り記号が正しい限り，その入力された時刻を使用します．入力されたパラメータが正しくない場合は，MSX-DOSはつぎのメッセージを表示します．

Invalid time

Enter new time：_

このメッセージは正しい入力がされるまで繰り返されます．

TYPE

タイプ

● **機　能**　**コンソールスクリーン上にファイルの内容を表示する.**

● **書　式**　**TYPE␣<ファイルスペック>**

● **解　説**　このコマンドは，ファイルを修正せずにその内容を調べる際に使用します.ファイルの名前を探すにはDIRを使用します.

TYPEコマンドは，タブコードをつぎのタブストップ(8カラムごとにタブストップが設定されている)までのスペースコードに展開します.

バイナリファイル(.COMなど)の内容をTYPEコマンドで表示すると，ベルコード，フォームフィードコード，およびエスケープシーケンスを含むコントロールキャラクタが送られ，正しい表示がおこなわれません.

第６章

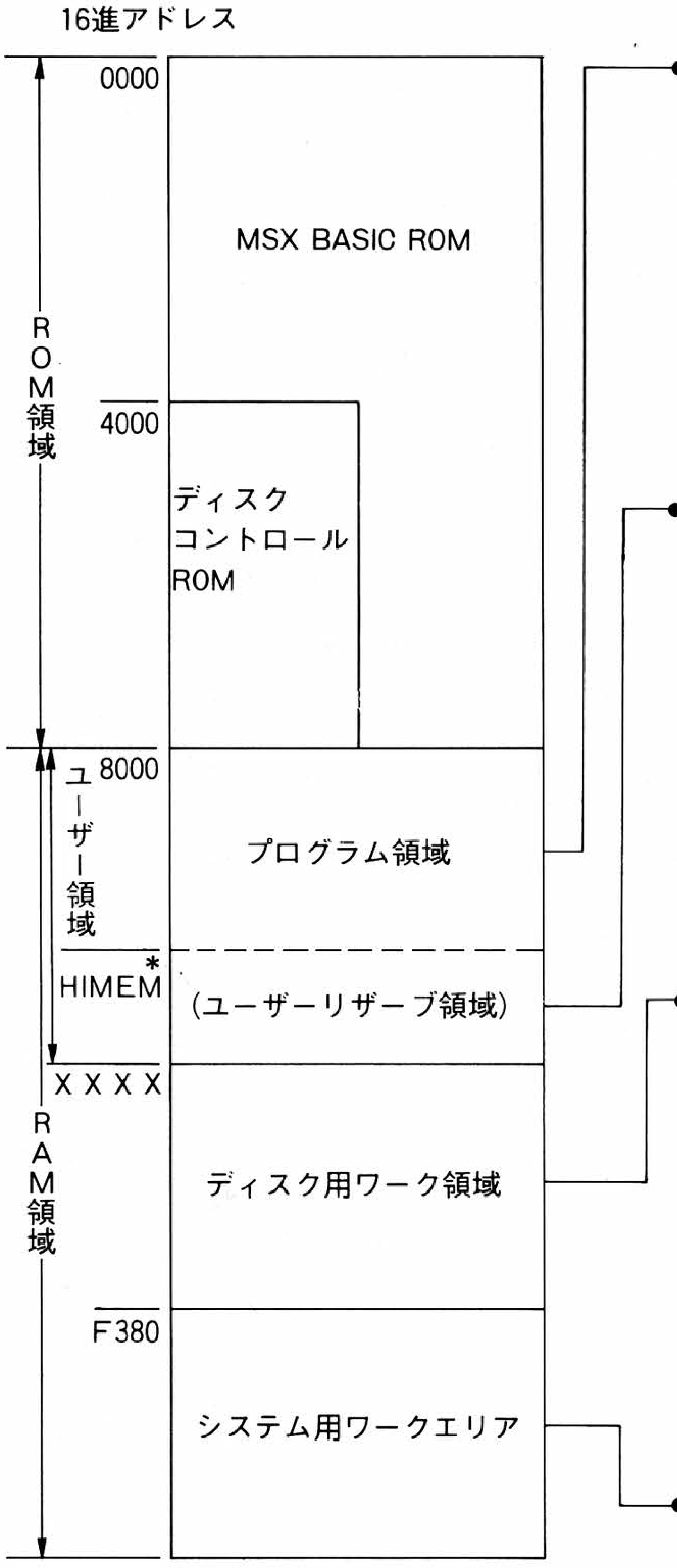
資 料 編

予約語表

ABS
AND
ASC
ATN
ATTR$
AUTO
BASE
BEEP
BIN$
BLOAD
BSAVE
CALL
CDBL
CHR$
CINT
CIRCLE
CLEAR
CLOAD
CLOSE
CLS
CMD
COLOR
CONT
COPY
COS
CSAVE
CSNG
CSRLIN
CVD
CVI
CVS
DATA
DEF
DEFDBL
DEFINT
DEFSNG
DEFSTR
DELETE
DIM
DRAW
DSKF
DSKI$
DSKO$
ELSE
END
EOF
EQV
ERASE
ERL
ERR
ERROR
EXP
FIELD
FILES
FIX
FN
FOR
FPOS
FRE
GET
GO TO
GOSUB
GOTO
HEX$
IF
IMP
INKEY$
INP
INPUT
INSTR
INT
IPL
KEY
KILL
LEFT$
LEN
LET
LFILES
LINE
LIST
LLIST
LOAD
LOC
LOCATE
LOF
LOG
LPOS
LPRINT
LSET
MAX
MERGE
MID$
MKD$
MKI$
MKS$
MOD
MOTOR
NAME
NEW
NEXT
NOT
OCT$
OFF
ON
OPEN
OR
OUT
PAD
PAINT
PDL
PEEK
PLAY
POINT
POKE
POS
PRESET
PRINT
PSET
PUT
READ
REM
RENUM
RESTORE
RESUME
RETURN
RIGHT$
RND
RSET
RUN
SAVE
SCREEN
SET
SGN
SIN
SOUND
SPACE$
SPC(
SPRITE
SQR
STEP
STICK
STOP
STR$
STRIG
STRING$
SWAP
TAB(
TAN
THEN
TIME
TO
TROFF
TRON
USING
USR
VAL
VARPTR
VDP
VPEEK
VPOKE
WAIT
WIDTH
XOR

メモリマップ

■DISK BASICで使用している場合



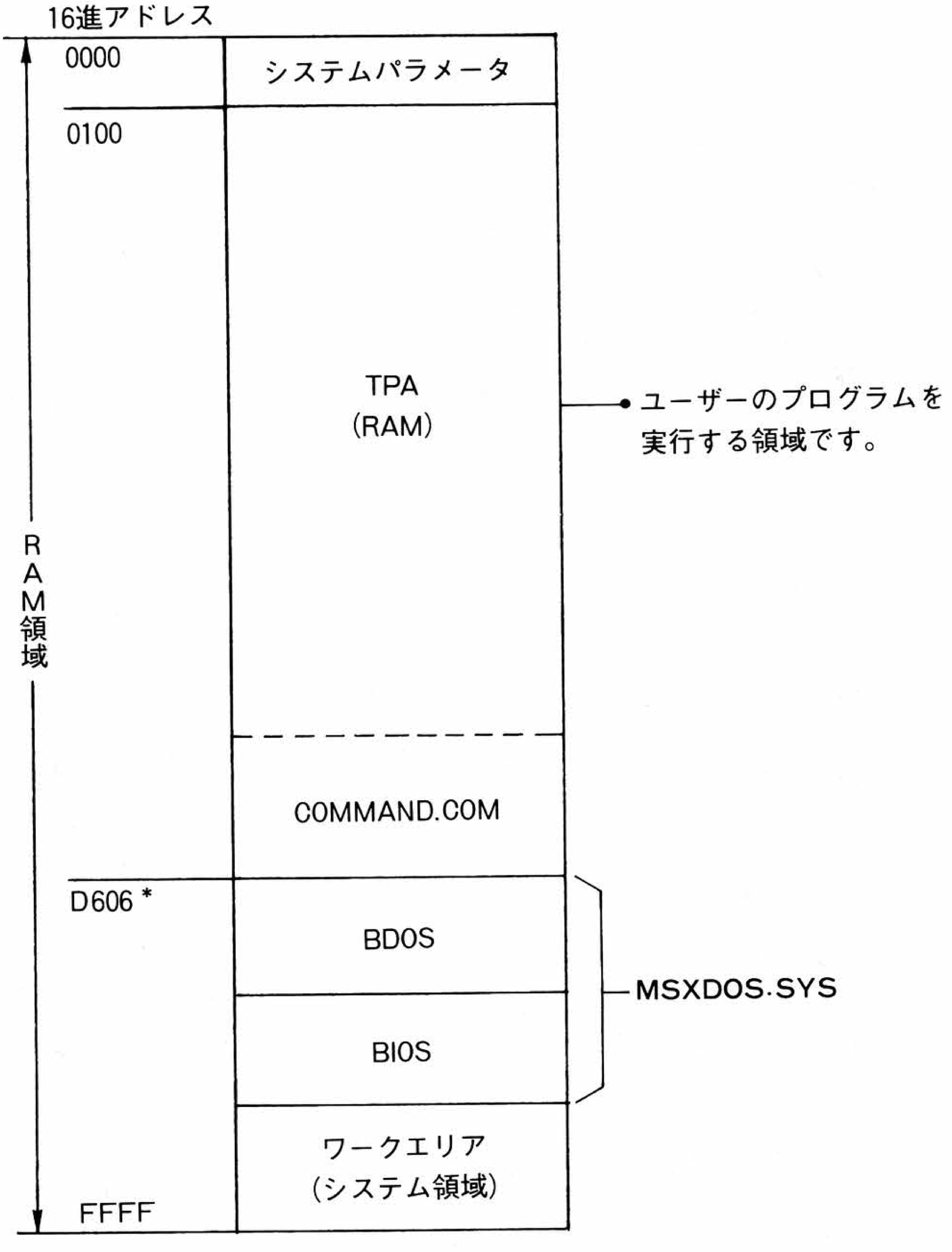
● BASICプログラムが格納される領域です。
プログラムに付随する変数領域、文字列領域、FOR～NEXT文などのスタックエリア、フリーエリアなども含みます。
フリーエリアは、FRE関数で求めることができます。

● 機械語ルーチンなどのために使うことができる領域です。
CLEAR文でプログラム領域の上限アドレスを設定することにより設けることができます。
＊この領域の上限アドレスは、HIMEM（&HFC4A および &HFC4B番地）の内容により示されます。

● ディスク用の入出力バッファ、FATなどに使用される領域です。
XXXX の番地は電源オン時のHIMEM(&HFC4A,&HFC4B番地)の内容により示されます。
(CLEAR文によりプログラム領域の上限アドレスを設定した後は異なります。)

● BASICが使用する領域です。

■MSX-DOSで使用している場合



＊ディスクのインターフェース基板が1枚の場合.

エラーメッセージ表 (DISK BASIC)

メッセージ	エラーコード	説　　明
Bad allocation table	60	ディスクがイニシャライズされていない.
Bad drive name	62	ドライブ名の指定が間違っている.
Bad file mode	61	シーケンシャルファイルに対してPUT文, GET文またはLOF関数が使用されたり, ランダムファイルに対してLOAD文が使用された. また, OPEN文のファイルモードとしてI, O, A, R以外が設定されている.
Bad file name	56	ファイル名が適当でない.
Bad file number	52	オープンしていない(開いていない)ファイル番号を指定した. 指定したファイル番号がMAXFILES文で指定した数を越えている.
Can't continue	17	CONTコマンドでプログラムの実行を再開できない. エラーが発生して終了したプログラムは再開できません. また, プログラムの1部を書き換えたり, CLEAR文などを実行すると, 次に実行すべきポインタが壊れてプログラムを再開することはできなくなります.
Device I/O error	19	カセットレコーダ, プリンタ, CRTなどへの入出力中にエラーが発生した.
Direct statement	57	アスキー形式のプログラムファイルをロード中, プログラム以外のデータがあった.
Disk full	66	ディスク上に空領域が無い.
Disk I/O error	69	ディスク入出力操作において, オペレーティングシステムが回復できない入出力エラーが生じた.
Disk offline	70	フロッピーディスクが装着されていない. 装着したフロッピーディスクがフォーマットされていない.
Disk write protected	68	書き込み禁止のディスクに書き込みを行った.
Division by zero	11	0による除算を行おうとした. または, 0に対して負のべき乗を行った.

メッセージ	エラーコード	説　　明
FIELD overflow	50	FIELD文で定義したフィールドサイズの合計が256バイトまたは指定の値を越えた.
File already exists	65	NAME文の新しいファイル名として指定された名前が, ディスク上に登録されているファイル名と重複した.
File already open	54	OPEN文で指定したファイル番号は, 他のファイルで使用中です.
File not found	53	指定されたファイルが見つからない.
File not OPEN	59	OPEN文で開いていないファイルに対して入出力を行おうとした.
File still open	64	ファイルのCLOSEを忘れている.
Illegal direct	12	ダイレクトモードでは実行できないステートメントをダイレクトモードで実行しようとした.
Illegal function call	5	ステートメントや関数の使い方が誤っている. 引数などが指定可能な範囲を越えている.
Input past end	55	ファイル中のデータをすべて読み込んだ後, さらに入力文を実行した. ファイル中のデータをすべて読み込んだかどうかはEOF関数で調べてください.
Internal error	51	BASIC内部でエラーが発生した.
Line buffer overflow	25	入力されたデータが許されている文字数を越えた.
Missing operand	24	文中に必要なパラメータが指定してない.
NEXT without FOR	1	NEXT文に対応するFOR文がない.
No RESUME	21	エラー処理ルーチンにRESUME文がない.

メッセージ	エラーコード	説　　明
Out of DATA	4	READ文で読み出せるデータがない．
Out of memory	7	プログラムが長すぎたり，配列が大きすぎたりなどの原因でメモリが足りなくなった．
Out of string space	14	文字列変数用のメモリが足りなくなった．
Overflow	6	演算結果や，入力された数値が型に応じた許容範囲を超えた．
Redimensioned array	10	配列変数を2重に定義しようとした．または，配列変数に添字の最大値10が割当てられた後で配列を定義しようとした．
Rename across disk	71	異なるドライブ間でNAMEを実行した．
RESUME without error	22	エラー処理ルーチン以外でRESUME文を使用した．
RETURN without GOSUB	3	GOSUB文とRETURN文が正しく対応していない．
Sequential I/O only	58	シーケンシャルファイルに対してランダム入出力を行おうとした．
String formula too complex	16	指定した文字列が複雑すぎる．文字式を2つ以上に分けてください．
String too long	15	文字列が255文字より大きい．
Subscript out of range	9	指定した配列変数の添字の値が，DIM文で定義した配列変数の添字の最大値より大きい．
Syntax error	2	文の書き方(構文)を誤っている.(かっこが対応していない．コマンド・ステートメントの綴りを誤っている．区切り記号がないなど)

メッセージ	エラーコード	説　　明
Too many files	67	ファイルの数が112を越えた．すなわち，ディレクトリの112個のエントリすべてが使用されている時に新しいファイルを生成しようとした．
Type mismatch	13	数式や数値関数に文字列を使ったり，文字式や文字関数に数値を使うなど，データの形式が一致しない．
Undefined line number	8	GOTO文，GOSUB文，IF文などで指定した行番号がプログラム中にない．
Undefined user function	18	DEF FNで定義していないユーザー定義関数を使っている．
Unprintable error	23 26〜49 72〜255	メッセージが定義されていないエラー．
Verify error	20	カセットにセーブされたプログラムはメモリ上のプログラムと内容が異なる． プログラムのセーブをやり直してください．

エラーメッセージ表(MSX-DOS)

メッセージ	説　　明
Bad command or file name	コマンド名かプログラム名が間違っている.
File creation error	新しいファイルの作成に失敗しました.
File not found	ファイルが見つからない.
Insufficient disk space	ディスク容量の空きがない.
Invalid date	日付が間違っている.
Invalid drive specification	ドライブ名の指定が間違っている.
Invalid parameter	パラメータが間違っている.
Invalid time	時間が間違っている.
Program too big to fit in memory	プログラムが大きくてメモリーに入りきらない.
Rename error	名前の変更に失敗しました.
Write error	書き込みに失敗しました.

索引

松下電器産業株式会社
情報機器部
〒571 大阪府門真市大字門真1006
電話 (06) 908-1151 (代表)

DFQF2006ZF

F0285M-4066